

Bayesian Spatial Binary Classification

Candace Berrett^{*1} and Catherine A. Calder^{†2}

¹*Department of Statistics, Brigham Young University, Provo, UT, USA*

²*Department of Statistics, The Ohio State University, Columbus, OH, USA*

January 11, 2016

Abstract

In analyses of spatially-referenced data, researchers often have one of two goals: to quantify relationships between a response variable and covariates while accounting for residual spatial dependence or to predict the value of a response variable at unobserved locations. In this second case, when the response variable is categorical, prediction can be viewed as a classification problem. Many classification methods either ignore response-variable/covariate relationships and rely only on spatially proximate observations for classification, or they ignore spatial dependence and use only the covariates for classification. The Bayesian spatial generalized linear (mixed) model offers a tool to accommodate both spatial and covariate sources of information in classification problems. In this paper, we formally define spatial classification rules based on these models. We also take a close look at two of these models that have been proposed in the literature, namely the probit versions of the spatial generalized linear model (SGLM) and the Bayesian spatial generalized linear mixed model (SGLMM). We describe the implications of the seemingly slight differences between these models for spatial classification and explore the issue of robustness to model misspecification through a simulation study. We also provide an overview of alternatives to the SGLM/SGLMM-based classifiers and illustrate the various methods using satellite-derived land cover data from Southeast Asia.

Keywords: generalized linear model, land cover, mixed models, MCMC, probit regression

1 Introduction

Prediction of unobserved binary or categorical variables can be cast as a classification problem, where a classification rule is used to assign an unobserved variable to a class, or category, based on a collection of observed *inputs* (e.g., predictors or covariate information). A classification rule is determined by a *decision function*, or a function of the inputs, which can be derived from either an underlying statistical model (e.g., logistic regression and discriminant analysis) or an algorithmic method such as support vector machines (SVM) and k-nearest neighbors (kNN) (see Hastie et al., 2001, for an overview). In this paper, we consider the spatial classification problem. That is, we seek to define classification rules to assign an unobserved variable associated with a spatial location (a particular point in a continuously-indexed spatial domain or an area in a discretely-indexed spatial domain) to one or more discrete classes. We refer to this spatial location as the *focal location* and

^{*}cberrett@stat.byu.edu

[†]calder@stat.osu.edu

the area surrounding it as the *neighborhood* of the focal location.

In classification problems involving spatially-referenced variables, often neighboring values of the unobserved/unknown variable should be used as inputs to the decision function, along with other inputs associated with the focal location and its neighbors. For example, binary or categorical images derived from satellite remote sensing often contain unobserved locations (or, “pixels” in this setting) due to errors in processing the raw data or measurement complications such as cloud cover. In these situations, formal classification methods are needed to assign values to the unobserved location so that the images can be used for various purposes in scientific investigations. While values of inputs associated with the focal location (e.g., land cover) may contain valuable information, knowledge of the class of neighboring locations may also be useful in classifying the focal location correctly. As we will illustrate, classification rules that rely on neighboring observations can be derived from the Bayesian spatial generalized linear and generalized linear mixed models (SGLMs and SGLMMs, respectively).

Some spatial classification methods have been proposed in the literature, many of which were motivated by remote sensing applications where the measured spectra serve as inputs/covariates. Unlike the classifiers derived from SGLM/SGLMMs in which spatial proximity explains the patterning of a categorical outcome after accounting for covariates (i.e., spatial dependence is in the ‘residuals’), these alternative spatial classifiers take advantage of spatial dependence in the covariates associated with each location. For example, Switzer (1980) and Mardia (1984) build on the traditional linear discriminant analysis by augmenting the covariates associated with the focal location with the covariates of neighboring locations in determining classification rules. Building on this idea, Šaltytė Benth and Dučinskas (2005) and Batsidis and Zografos (2011) explicitly model the strength of spatial dependence in the covariates to guide the selection of the spatial extent and weighting of neighboring covariate values to be used in the classification rule. In cases where existing spatial classification methods do make use of the class of neighboring locations (Klein and Press, 1992; Press, 1996), the dependence is not directly modeled. Instead, the classes of neighboring locations are used to select the spatial extent of neighboring covariate values that are used as inputs to the classification rule. These final two methods, to our knowledge, are also the only existing Bayesian spatial classifiers. These spatial approaches to classification have clear utility in remote sensing applications when entire scenes (images) are completely unobserved and spectra (covariates) are often strongly informative and exhibit strong spatial dependence. However, in cases where only some pixel classes are missing, SGLM/SGLMM-based classifiers, which make direct use of both neighboring class information and covariates, is desirable.

The primary goals of this paper are to formally define spatial binary classifiers based on the probit versions of the SGLM and SGLMM and compare these two classifiers in terms of the complexity of the underlying model and the robustness to misspecification of the underlying model. As we discuss below, the probit SGLM and SGLMM have been used in the literature to model spatially-dependent binary data. However, we are not aware of existing studies exploring model misspecification in the classification/prediction setting and thus this discussion is the primary contribution of this paper. The formalism we introduce to define spatial binary classifiers based on the SGLM/SGLMM allows us to readily compare the performance of these methods to other spatial and non-spatial binary classifiers, a secondary goal of the paper. We provide a comprehensive review of these alternative methods in an appendix (Appendix A).

2 SGLM and SGLMM Classifiers

2.1 Spatial Generalized Linear (Mixed) Models

Following the seminal work of Diggle et al. (1998), the generalized linear mixed model (GLMM) has become the go-to framework for modeling spatially-dependent phenomena for which Gaussian distributional assumptions are inappropriate. Through the introduction of spatially-dependent random effects within a generalized linear model (McCullagh and Nelder, 1989), standard models for both continuous and discretely-indexed Gaussian data (e.g., Gaussian processes, spatial autoregressive models) can be readily adapted to the non-Gaussian data situation. (See Paciorek, 2007, for several examples, and Hughes and Haran, 2013, for a more recent approach to dimension reduction and alleviation of confounding in these models.) We review the SGLMM for dichotomous spatial data below in order to make connections between specific models that have appeared in the literature and to provide a framework for defining spatial classification methods.

Let $\mathbf{Y} = (Y_1, \dots, Y_n)'$ be an observable binary response variables associated with locations $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_n)'$, which are a subset of either a continuous or discrete spatial domain. The standard generalized linear model (GLM) is specified through three components: the random component, the link function, and the systematic component. The SGLMM can be written using these components as well. For the random component of the model, we assume that conditional on unknown parameters $\boldsymbol{\beta}$ and $\boldsymbol{\nu}$, the Y_i 's are independent and that

$$Y_i | \boldsymbol{\beta}, \boldsymbol{\nu} \sim \text{Bin}(1, p_i), \quad (1)$$

where $p_i = P(Y_i = 1 | \boldsymbol{\beta}, \boldsymbol{\nu})$. For the link function, we let

$$g(p_i) \equiv \eta_i, \quad (2)$$

where generally we assume $g(\cdot)$ is a one-to-one, monotone, continuous, and differentiable function. For binary response variables, common link functions are the logit and probit functions, $g(p_i) = \log(p_i/(1 - p_i))$ and $g(p_i) = \Phi^{-1}(p_i)$, respectively, where $\Phi(\cdot)$ is the standard normal cumulative distribution function and $\Phi^{-1}(\cdot)$ denotes its inverse. Finally, the systematic component of the model is

$$\eta_i = \mathbf{x}_i' \boldsymbol{\beta} + \nu_i, \quad (3)$$

where \mathbf{x}_i is an $\ell \times 1$ vector of covariates measured at location \mathbf{s}_i , $\boldsymbol{\beta}$ is an $\ell \times 1$ vector of regression coefficients, and $\boldsymbol{\nu} = (\nu_1, \dots, \nu_n)'$ is a spatial random effects vector with components associated with each spatial location.

By definition, the inclusion of random effects in the systematic component of the model (3) makes the model a GLMM. It also allows for the introduction of spatial dependence through the dependence structure of $\boldsymbol{\nu}$. In particular, we take

$$\boldsymbol{\nu} | \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta})), \quad (4)$$

where $\mathbf{0}$ is an $n \times 1$ vector of zeros and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is the $n \times n$ spatial covariance matrix parameterized by $\boldsymbol{\theta}$. (For example, the spatial covariance matrix could correspond to a Matérn covariance function

or a spatial autoregressive model.) As Banerjee et al. (2004) point out, $\Sigma(\theta)$ would not have a nugget, so that we can write

$$\Sigma(\theta) = \theta_1 K(\theta_2), \quad (5)$$

where $\theta = (\theta_1, \theta_2)'$ and $K(\theta_2)$ is an $n \times n$ spatial dependence matrix parameterized by θ_2 . Although not always the case, θ_1 is often taken to be the common variance parameter among the ν_i s and $K()$ is a spatial correlation function.

The probit link function is often used in GLMs/GLMMs in the Bayesian setting since the well-known latent-variable representation of the probit GLM facilitates model fitting via the Gibbs sampler (Albert and Chib, 1993). This same latent variable representation also suggests an alternative GLM for dichotomous spatial data (compared to the spatial GLMM described above) that has been used in the literature, so we briefly review it here. Introducing a collection of latent variables $\mathbf{Z} = (Z_1, \dots, Z_n)'$ associated with each spatial location, we take

$$Y_i = \begin{cases} 1, & Z_i \geq 0, \\ 0, & Z_i < 0 \end{cases}$$

where

$$Z_i = \mathbf{x}_i' \boldsymbol{\beta} + \nu_i + \epsilon_i, \quad (6)$$

$\epsilon_i \stackrel{iid}{\sim} \mathcal{N}(0, 1)$, and, as above, $\boldsymbol{\nu} | \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \Sigma(\boldsymbol{\theta}))$. We can see that this is simply an alternative representation of the probit SGLMM described previously since

$$p_i = P(Y_i = 1 | \boldsymbol{\beta}, \boldsymbol{\nu}) = P(Z_i \geq 0 | \boldsymbol{\beta}, \boldsymbol{\nu}) = \Phi(\mathbf{x}_i' \boldsymbol{\beta} + \nu_i).$$

Notice, however, that

$$(\boldsymbol{\nu} + \boldsymbol{\epsilon}) | \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \Sigma^*(\boldsymbol{\theta})), \quad (7)$$

where $\Sigma^*(\boldsymbol{\theta}) = \mathbf{I} + \Sigma(\boldsymbol{\theta})$ and \mathbf{I} is the n -dimensional identity matrix. If $\Sigma^*(\boldsymbol{\theta})$ is equal to $\Sigma(\boldsymbol{\theta})$ instead (i.e., dropping the identity matrix), we have a spatial GLM (SGLM) as opposed to an SGLMM. That is, there are no random effects in the systematic component of the GLM since the ν_i s now take the place of the random component of the GLM in the latent variable representation of the model. For this SGLM, we write

$$Z_i = \mathbf{x}_i' \boldsymbol{\beta} + \nu_i. \quad (8)$$

With this specification, the p_i s are no longer found using the traditional inverse probit link function (the independent standard normal distribution). Instead, the joint probabilities can be found using the multivariate normal distribution.

Both the probit SGLMM and probit SGLM have been used in the literature. Following De Oliveira (2000), Higgs and Hoeting (2010) and Berrett and Calder (2012) considered the SGLM, which is sometimes referred to as a clipped Gaussian process. More recent work by Schliep and Hoeting (2013) makes use of the SGLMM for the ordered-category response variable case. Given that both models are used in the literature, we briefly discuss the implications of their seemingly slight differences. Comparing the probit SGLMM and SGLM described above, we note that while the Y_i s are conditionally independent under the SGLMM since the dependence is introduced through

the dependence structure of the latent ν_i s (1), the Y_i s are not conditionally independent in the SGLM. To see this, consider the *conditional* likelihood function, $L(\boldsymbol{\beta}, \boldsymbol{\nu}) = P(\mathbf{Y} = \mathbf{y} | \boldsymbol{\beta}, \boldsymbol{\nu})$, for the SGLMM, which can be factored into n components:

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\nu}) &= \prod_{i=1}^n (\Phi(\mathbf{x}'_i \boldsymbol{\beta} + \nu_i) I(y_i = 1) + (1 - \Phi(\mathbf{x}'_i \boldsymbol{\beta} + \nu_i)) I(y_i = 0)) \\ &= \prod_{i=1}^n \int_{\mathcal{A}_i} \phi(z_i; \mathbf{x}'_i \boldsymbol{\beta} + \nu_i, 1) dz_i, \end{aligned}$$

where $\phi(\cdot; m, v^2)$ is the density function of the univariate normal distribution with mean m and variance v^2 and

$$\mathcal{A}_i = \begin{cases} (-\infty, 0), & \text{if } y_i = 0, \\ [0, \infty), & \text{if } y_i = 1. \end{cases}$$

For the spatial GLM where there are no random effects in the systematic component of the model, the likelihood, $L(\boldsymbol{\beta}, \boldsymbol{\theta}) = P(\mathbf{Y} = \mathbf{y} | \boldsymbol{\beta}, \boldsymbol{\theta})$, is the integral over a multivariate normal distribution:

$$P(\mathbf{Y} = \mathbf{y} | \boldsymbol{\beta}, \boldsymbol{\theta}) = \int_{\mathcal{A}} \phi_n(\mathbf{Z}; \mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma}(\boldsymbol{\theta})) d\mathbf{Z},$$

where $\phi_n(\cdot, \mathbf{m}, \mathbf{V})$ is the density function of the n -dimensional multivariate normal distribution with mean \mathbf{m} and covariance \mathbf{V} ,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}'_1 \\ \vdots \\ \mathbf{x}'_n \end{bmatrix},$$

and $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$.

Since the primary focus of this paper is spatial classification and model comparison, we defer discussion of model fitting to Appendix B.1. However, we mention here recent approaches by Hughes and Haran (2013) and Hanks et al. (2015) to fitting a low-rank SGLMM for areal and geostatistical data, respectively, in the presence of spatial confounding. It is well-known that when covariates are spatially dependent that coefficient estimates can be biased and variances inflated (e.g., Reich et al., 2006; Hughes and Haran, 2013; Hanks et al., 2015). However, the impact of spatial confounding on prediction of the response variable has not been explored. While examining this more fully is beyond the scope of this paper, we do compare the SGLM- and SGLMM-based classifiers' performance when the covariates are spatially dependent in Section 3.2.2. We discuss the Hughes and Haran (2013) parameterization of the SGLMM more fully in Appendix B.2 and make use of their approach as an additional SGLMM classifier in Section 4.

While the probit SGLMM and SGLM described above are nearly identical – they differ only in terms of the inclusion of a identity matrix component in the model for $\boldsymbol{\Sigma}^*(\boldsymbol{\theta})$ – it is not immediately clear how the models compare in terms of their predictive performance. The difference between the models is reminiscent of a nugget effect, which is frequently used in geostatistical models and its inclusion is known to influence both point predictions and corresponding uncertainty statements. Before conducting a comparison of the two models, we first describe how these models can be used to define spatial classifiers.

2.2 Model-Based Classification

In a modeling framework, imputing unknown response variables is often viewed as a prediction problem. In the case where the unknown variable is categorical, imputation is often viewed as a classification problem and many model-based and algorithmic classification methods exist. To better relate the prediction methods from the SGLMM and SGLM to the other classification methods and to define a common classification rule among the various methods, we describe and build on the traditional classification problem to define decision functions for the SGLM and SGLMM.

2.2.1 The Classification Problem

In the canonical classification problem, we have a collection of paired observations $\{(y_i, \mathbf{x}_i); i = 1, \dots, n\}$, where y_i is the observed value of a binary response variable Y_i and \mathbf{x}_i is an $\ell \times 1$ vector of covariates used as inputs in determining decision function boundaries for the classes. The binary response variable is an indicator identifying the class to which the set of observed inputs belong. Although many classification methods can be generalized to the multi-category/class setting, we restrict our description of these methods to the binary setting in which there are two classes, C_0 and C_1 , where C_j represents the class of observations where $Y = j$. Of the observations $\{(Y_i, \mathbf{x}_i); i = 1, \dots, n\}$, n_0 fall into class C_0 and n_1 fall into class C_1 , and $n_0 + n_1 = n$.

A classification method defines a decision function $\delta(\boldsymbol{\omega})$, where $\boldsymbol{\omega}$ is the set of applicable inputs, model parameters, and in the spatial setting, surrounding observations. Based on this decision function, we can define a classification rule for an unobserved Y that we denote generically by Y^0 with corresponding covariate information \mathbf{x}^0 . As described in Hastie et al. (2001), the Bayes classifier for a 0-1 loss function is

$$y_{pred}^0 = \arg \max_j P(C_j | \boldsymbol{\omega}), \quad (9)$$

where $P(C_j | \boldsymbol{\omega}) = p_j(\boldsymbol{\omega})$ is the conditional probability of class C_j given the inputs $\boldsymbol{\omega}$. In other words, the optimal classification of Y^0 is the most likely class given $\boldsymbol{\omega}$. In the two-category case, we can rewrite (9) as

$$y_{pred}^0 = \begin{cases} 1, & \text{if } \delta(\boldsymbol{\omega}) > 1 \\ 0, & \text{otherwise} \end{cases}. \quad (10)$$

for $\delta(\boldsymbol{\omega}) = p_1(\boldsymbol{\omega})/p_0(\boldsymbol{\omega})$.

For model-based classification, the conditional probabilities (and, hence decision functions) depend on unknown parameters denoted by $\boldsymbol{\xi}$, which must be estimated from the observed data. We write $\boldsymbol{\omega}_{\boldsymbol{\xi}}$ to explicitly capture the inclusion of the parameters in the collection of inputs. In a frequentist setting, the unknown parameters, $\boldsymbol{\xi}$, are often set equal to the maximum likelihood estimates (MLE), so that

$$\hat{p}_j(\boldsymbol{\omega}) \equiv p_j(\boldsymbol{\omega}_{\hat{\boldsymbol{\xi}}}),$$

where $\hat{\boldsymbol{\xi}}$ is the MLE of $\boldsymbol{\xi}$. We refer to the classification rule given by (10) with $\delta(\boldsymbol{\omega})$ set equal to $\hat{\delta}(\boldsymbol{\omega}) = \hat{p}_1(\boldsymbol{\omega})/\hat{p}_0(\boldsymbol{\omega})$ as the *maximum likelihood classifier*. Alternatively, there are two Bayesian estimation approaches we consider. First, we can set $\boldsymbol{\xi}$ equal to its posterior mean,

$$\tilde{p}_j(\boldsymbol{\omega}) \equiv p_j(\boldsymbol{\omega}_{\tilde{\boldsymbol{\xi}}}),$$

where $\tilde{\boldsymbol{\xi}} = E[\boldsymbol{\xi}|\mathbf{y}]$. In this case, the classification rule is given by (10) with $\delta(\boldsymbol{\omega})$ replaced by $\tilde{\delta}(\boldsymbol{\omega}) = \tilde{p}_1(\boldsymbol{\omega})/\tilde{p}_0(\boldsymbol{\omega})$. This classification rule can be viewed as the Bayesian analog of the maximum likelihood classifier, and we refer to it as the *posterior mean classifier*. An alternative Bayesian approach is to marginalize over the posterior distribution of $\boldsymbol{\xi}$ and take

$$\bar{p}_j(\boldsymbol{\omega}) \equiv \int_{\Xi} p_j(\boldsymbol{\omega}_{\boldsymbol{\xi}}) \pi(\boldsymbol{\xi}|\mathbf{y}) d\boldsymbol{\xi}, \quad (11)$$

where $\pi(\boldsymbol{\xi}|\mathbf{y})$ is the posterior distribution of $\boldsymbol{\xi}$ given \mathbf{y} defined on the parameter space Ξ . As before, we let $\bar{\delta}(\boldsymbol{\omega}) = \bar{p}_1(\boldsymbol{\omega})/\bar{p}_0(\boldsymbol{\omega})$ replace $\delta(\boldsymbol{\omega})$ in (10) and refer to the resulting classification rule as the *posterior predictive classifier*.

In practice, we evaluate the integral in (11) via Monte Carlo integration. For example, for the Bayesian non-spatial GLM, we approximate $\bar{p}_1(\boldsymbol{\omega})$ by drawing a realization of $Y^{0[t]} \sim \text{Bernoulli}(p_1(\mathbf{x}^0, \boldsymbol{\beta}^{[t]}))$ for $t = 1, \dots, T$ and setting $\bar{p}_1(\boldsymbol{\omega}) = \sum_{t=1}^T I(Y^{0[t]} = 1)/T$, where $\boldsymbol{\beta}^{[t]}$ are the draws from the posterior distribution of $\boldsymbol{\beta}$, and T is the number of draws from the posterior distribution. In the following subsections, our focus is on defining decision functions as it provides a unifying framework for both the model-based classifiers discussed below and the alternative classifiers discussed in Appendix A.

2.2.2 Spatial Classification

While the decision functions based on the non-spatial GLM's rely only on covariates and regression coefficients (see Appendix A.1 for details), for the spatial case, the decision function $\delta(\boldsymbol{\omega})$ also depends on the categories of the surrounding observations. Because of this dependence, it is necessary to define a joint distribution for the latent variables introduced in equation (6) for both the observations and the focal location. Note that we generally define the same decision functions for the probit SGLMM and SGLM, since marginally, these models only differ in the definition of $\boldsymbol{\Sigma}^*(\boldsymbol{\theta})$ in (7).

Let $\mathbf{Z}^0 = (Z^0, \mathbf{Z}')'$ be the $(n+1) \times 1$ vector of latent variables, where Z^0 is the latent variable for the unobserved location and \mathbf{Z} is the $n \times 1$ vector of latent variables for the observations. Then, combining equations (6) and (7),

$$\mathbf{Z}^0 \sim \mathcal{N}(\mathbf{X}^0 \boldsymbol{\beta}, \boldsymbol{\Sigma}^{*0}(\boldsymbol{\theta})), \quad (12)$$

where $\mathbf{X}^0 = (\mathbf{x}^0, \mathbf{X}')'$,

$$\boldsymbol{\Sigma}^{*0} = \begin{bmatrix} \sigma^0(\boldsymbol{\theta}) & \boldsymbol{\sigma}(\boldsymbol{\theta})' \\ \boldsymbol{\sigma}(\boldsymbol{\theta}) & \boldsymbol{\Sigma}^*(\boldsymbol{\theta}) \end{bmatrix},$$

$\sigma^0(\boldsymbol{\theta})$ is the variance of Z^0 , and $\boldsymbol{\sigma}(\boldsymbol{\theta})$ is the $n \times 1$ vector representing the covariance of Z^0 and \mathbf{Z} . It follows that the distribution for the latent variable at an unobserved location is

$$Z^0 | \mathbf{x}^0, \mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{Z} \sim \mathcal{N}(\mu_{Z^0}, \sigma_{Z^0})$$

where

$$\mu_{Z^0} = \mathbf{x}^{0'} \boldsymbol{\beta} + \boldsymbol{\sigma}(\boldsymbol{\theta})' (\boldsymbol{\Sigma}^*(\boldsymbol{\theta}))^{-1} (\mathbf{Z} - \mathbf{X}' \boldsymbol{\beta}), \quad (13)$$

$$\sigma_{Z^0}^2 = \sigma^0(\boldsymbol{\theta}) - \boldsymbol{\sigma}(\boldsymbol{\theta})' (\boldsymbol{\Sigma}^*(\boldsymbol{\theta}))^{-1} \boldsymbol{\sigma}(\boldsymbol{\theta}). \quad (14)$$

We can easily include sampling from this distribution in the first step of the MCMC algorithm so that we can obtain draws from the posterior distribution of Z^0 . In this spatially-dependent case, $\omega = (\mathbf{x}^0, \mathbf{X}, \beta, \theta, \mathbf{Z})$ and

$$\begin{aligned} p_1(\omega) &\equiv p_1(\mathbf{x}^0, \mathbf{X}, \beta, \theta, \mathbf{Z}) \\ &= P(Z^0 > 0 | \mathbf{x}^0, \mathbf{X}, \beta, \theta, \mathbf{Z}) \\ &= \Phi \left(\frac{\mu_{Z^0}}{\sqrt{\sigma_{Z^0}^2}} \right). \end{aligned}$$

Thus, the decision function at \mathbf{x}^0 is

$$\delta_{SGLM}(\omega) \equiv \delta_{SGLM}(\mathbf{x}^0, \mathbf{X}, \beta, \theta, \mathbf{Z}) = \frac{P(Z^0 > 0 | \beta, \theta, \mathbf{Z})}{1 - P(Z^0 > 0 | \beta, \theta, \mathbf{Z})} = \frac{\Phi \left(\mu_{Z^0} / \sqrt{\sigma_{Z^0}^2} \right)}{1 - \Phi \left(\mu_{Z^0} / \sqrt{\sigma_{Z^0}^2} \right)}.$$

3 Comparison of the SGLM and SGLMM Classifiers

3.1 A Unified Model and Illustrative Comparisons

To facilitate comparison of the SGLM and SGLMM classifiers, we consider the following representation of $\Sigma^*(\theta)$ in (7) of the SGLMM. Recall that $\Sigma^*(\theta) = \mathbf{I} + \Sigma(\theta) = \mathbf{I} + \theta_1 K(\theta_2)$. As discussed in Appendix B.1, we can augment the parameter space with an additional non-identifiable multiplicative factor on $\Sigma^*(\theta)$. This additional parameter facilitates MCMC mixing and allows us to consider a continuum of models indexed by a parameter κ . Specifically, we write

$$\Sigma^*(\theta) = \gamma^2((1 - \kappa)\mathbf{I} + \kappa\Sigma(\theta)), \quad (15)$$

where γ^2 is a non-identifiable “working” parameter and $\kappa \in [0, 1]$ is the proportion of the marginal variance associated with the spatially-dependent variance component. In other words, $\kappa\gamma^2 = \theta_1$. The SGLM can now readily be viewed as a special case of the SGLMM. Using this parameterization, as $\kappa \rightarrow 1$, the SGLMM becomes the SGLM and, as $\kappa \rightarrow 0$, the SGLMM becomes the independent probit model.

Although the SGLM and SGLMM are similar, data generated from these models can differ substantially in terms of the spatial structure. For example, Figure 1 shows data sets generated under different SGLMMs, where for a fixed value of θ_2 the data generated vary from completely independent to maximally spatially dependent (SGLM).

For spatially-dependent binary variables, we expect neighboring observations to belong to the same class and wish to exploit this feature for improved classification. To examine how the SGLM and SGLMM classifiers allow for this property, we provide two illustrations. For both illustrations, we consider the SGLMM with $\Sigma^*(\theta)$ given by (15) and let $K(\theta_2)$ be defined by a conditionally autoregressive (CAR) spatial dependence structure (Banerjee et al., 2004) parameterized by ρ . That is, $K(\theta_2) = (\mathbf{I} - \rho\mathbf{W})^{-1}$, where $\theta_2 = \rho$ and \mathbf{W} is a fixed, row-standardized $n \times n$ spatial

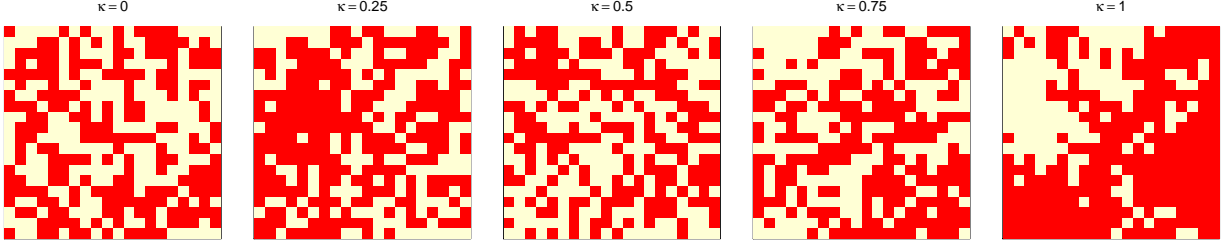


Figure 1: Examples of binary data generated from the SGLM/SGLMM, where each data set is generated using a constant zero mean and varying levels of dependence. The left is generated from the least spatially dependent model (i.e., independent probit) and the far right is generated from the most spatially dependent model (SGLM).

neighborhood matrix. In the first illustration where $n = 2$,

$$\mathbf{W} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

First, consider the simple case where $n = 2$ so that we have observable binary variables Y_1 and Y_2 and focus on the event that both observations are from the same class (i.e., $\{Y_1 = 1, Y_2 = 1\}$ or $\{Y_1 = 0, Y_2 = 0\}$). The probability of this event can be obtained from the joint distribution of the latent variables Z_1 and Z_2 . The plot on the left in Figure 2 corresponds to the case when Z_1 and Z_2 are assumed to be conditionally independent given the values of other model parameters; this is the same as the standard independent probit regression model ($\kappa = 0$). The orange circle represents the 95 percent ellipse of the joint probability distribution (i.e., the volume under the joint probability distribution is 0.95), which in this case is the circular bivariate normal distribution. The center of this ellipse is marked by the orange point. Note that the dark orange and light orange shaded areas within this ellipse represent $P(\{Y_1 = 1, Y_2 = 1\})$ and $P(\{Y_1 = 0, Y_2 = 0\})$, respectively, again assuming other model parameters are fixed. This shaded ellipse is included in the remaining plots in Figure 2 to aid in making comparisons. The other images depict the same probabilities under different values of κ for a fixed ρ . These plots illustrate how as κ increases, so do the probabilities of $P(\{Y_1 = 1, Y_2 = 1\})$ and $P(\{Y_1 = 0, Y_2 = 0\})$, even though the “spatial dependence parameter,” ρ , is fixed. Therefore, for values of $0 < \kappa < 1$, the strength of spatial dependence (i.e., probability of neighbors belonging to the same class) in the SGLMM does not simply depend on the value of ρ .

In the second illustration, $n = 3$ and

$$\mathbf{W} = \begin{bmatrix} 0 & 1 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \end{bmatrix},$$

so that the location of Y_2 is a neighbor to the locations of Y_1 and Y_3 , but these two remaining locations are not neighbors of each other. Again we are interested in the probability that these observations are the same: $P(\{Y_1 = 0, Y_2 = 0, Y_3 = 0\})$ and $P(\{Y_1 = 1, Y_2 = 1, Y_3 = 1\})$. For

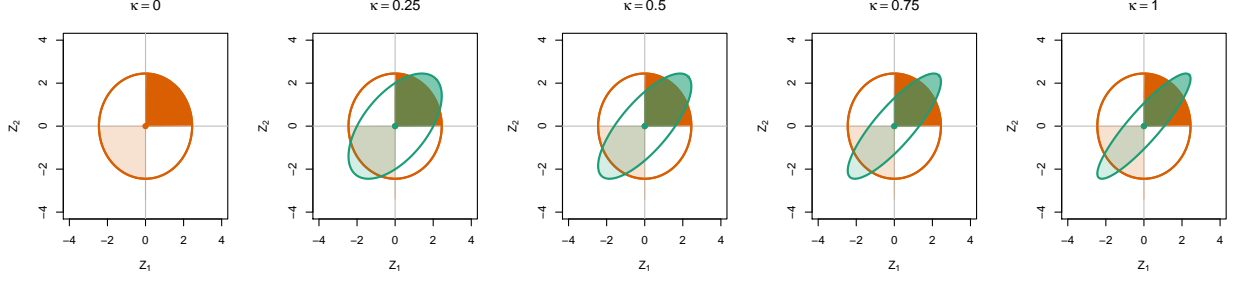


Figure 2: Examples of 95 percent probability ellipses for two latent variables, Z_1 and Z_2 , for the SGLM/SGLMM. The shaded areas represent the probability of Y_1 and Y_2 being in the same class.

shorthand, we represent this “sameness probability” by $P(\{Y_1 = Y_2 = Y_3\})$. The solid lines in Figure 3 show the value of this sameness probability for different combinations of ρ and κ when all other model parameters fixed. As expected, for larger values of ρ and κ , $P(\{Y_1 = Y_2 = Y_3\})$ increases. Notice, however, that there are several values of (ρ, κ) that result in equal values of $P(\{Y_1 = Y_2 = Y_3\})$. In fact, Figure 3 shows that $(0.935, 0.25)$, $(0.866, 0.5)$, $(0.790, 0.75)$, $(0.707, 1)$ all result in $P(\{Y_1 = Y_2 = Y_3\}) = 0.5$.

Now consider the probability that Y_3 is different than Y_1 and Y_2 : $P(\{Y_1 = Y_2 \neq Y_3\}) = P(\{Y_1 = 0, Y_2 = 0, Y_3 = 1\}) + P(\{Y_1 = 1, Y_2 = 1, Y_3 = 0\})$. The dotted lines in Figure 3 represent these probabilities for different values of ρ and κ , again with the other model parameters fixed. Although the above combinations of (ρ, κ) values resulted in equal values of $P(\{Y_1 = Y_2 = Y_3\})$, they do not result in the equal values of $P(\{Y_1 = Y_2 \neq Y_3\})$. As shown by the points in Figure 3, the values of $P(\{Y_1 = Y_2 \neq Y_3\})$ are close, but not equal. This suggests that these models, and consequently classification rules derived from them, are robust to misspecification; that is, if the data were generated under the SGLMM/SGLM, the incorrect model can correct for the model misspecification through adjusting values of ρ and/or κ . To explore this in a more realistic setting where $n > 3$, in the following subsection we conduct a simulation study and compare the classifications for the SGLMM/SGLM under model misspecification.

3.2 Robustness to Model Misspecification

To more fully examine the differences between the SGLM- and SGLMM-based classifiers, we examine error rates for both classifiers under different generative models for the data. Specifically, we are interested in determining if the model under which the data were generated performs the best in terms of classification, or if one model consistently outperforms the other. Before describing the simulation study, we first define the method for computing the misclassification error rates we use for comparison.

3.2.1 Error Rates

When developing and assessing classification methods, the observations are randomly divided into two subsets, the training and test data sets, of sizes n_{train} and n_{test} , respectively. This allows us to fit the model on which the classifier is based using the training data and evaluate the method’s

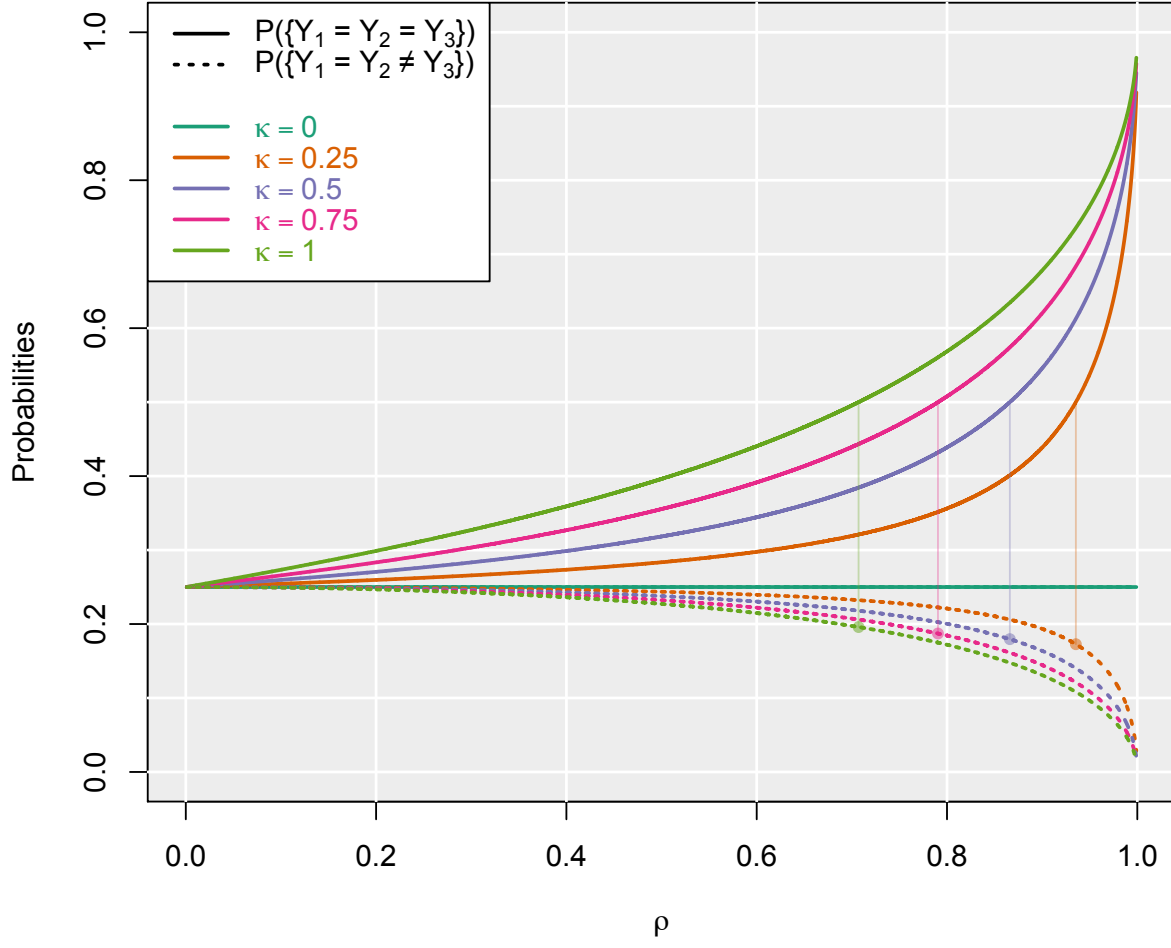


Figure 3: Considering different combinations of classes for three locations, this plot shows how their probabilities change for different values of ρ and κ . The thick solid lines correspond to the probability that all three locations belong to the same class. The dotted lines correspond to the probability that the third location belongs to a different class than the other two. The thin solid lines connect these two probabilities for equivalent values of (ρ, κ) .

out-of-sample predictive ability. When we compare classification methods, we consider training and test error rates, or the number of incorrect classifications among training and test data sets, divided by the sample size of each data set, respectively. We expect training error to decrease as model complexity increases. As our toy examples in the previous section illustrate, determining the relative complexity of the SGLM versus SGLMM is not straightforward – we cannot simply count the number of parameters and conclude that the SGLMM is a more complex version of the SGLM. On the other hand, test error rates allow us to explore potential overfitting.

In Section 2.2.1 we described three model-based classifiers derived from different approaches for estimating the decision function: maximum likelihood, posterior mean, and posterior predictive. Determining error rates using maximum likelihood and posterior mean classifiers is straightforward. However, for spatial classification, determining training error rates based on the posterior predictive classifier is not. Under the posterior predictive classifier, for the test data, we can simply use the latent Z_j , for $j = 1, \dots, n_{test}$, as sampled within the Gibbs sampler. However, for the training data, the latent Z_i , for $i = 1, \dots, n_{train}$, are sampled within the Gibbs sampler given the observed Y_i . In this case, to use the latent Z_i as inputs, we must sample these values as if the Y_i are unknown, otherwise we would have perfect prediction.

For the Bayesian spatial probit regression model, we rely on the observed surrounding observations to provide information about the category of the unobserved locations. Because of this, evaluating the training error is not straightforward. We propose the following two approaches for the spatial probit posterior predictive classifier:

A. One-at-a-Time Training Error:

- (i) Take samples $(\beta^{[t]}, \theta^{[t]}, \mathbf{Z}_{-i}^{[t]})$, for $t = 1, \dots, T$, where $\mathbf{Z}_{-i}^{[t]}$ is an $(n-1) \times 1$ vector of sampled Z_j for $j = 1, \dots, i-1, i+1, \dots, n_{train}$ and sample a corresponding $Z_i^{[t]} \sim N(\mu_{Z_i}, \sigma_{Z_i}^2)$ where

$$\begin{aligned}\hat{\mu}_{Z_i} &= \mathbf{x}'_i \beta^{[t]} + \Sigma^*(\theta^{[t]})_{i,-i} \left(\Sigma^*(\theta^{[t]})_{-i,-i} \right)^{-1} (\mathbf{Z}_{-i}^{[t]} - \mathbf{X}_{-i} \beta^{[t]}) \\ \hat{\sigma}_{Z_i}^2 &= \Sigma^*(\theta^{[t]})_{i,i} - \Sigma^*(\theta^{[t]})_{i,-i} \left(\Sigma^*(\theta^{[t]})_{-i,-i} \right)^{-1} \Sigma^*(\theta^{[t]})_{-i,i}\end{aligned}$$

and \mathbf{X}_{-i} is \mathbf{X} with the i^{th} row removed, and $\Sigma^*(\theta^{[t]})_{j,-k}$ is the j^{th} row of the estimated spatial correlation of \mathbf{Z} with the k^{th} column removed. (Note that $\mathbf{Z}_{-i}^{[t]}$ are the posterior samples obtained from the MCMC algorithm, however, the $Z_i^{[t]}$ s are not the same as those sampled in the MCMC algorithm.)

- (ii) Determine $\bar{p}_1(\omega) = \sum_{t=1}^T I(Z_i^{[t]} > 0)/T$ (suppressing the notation for the i -th observation in $\bar{p}_1(\omega)$), and let Y_i^0 be the predicted value of Y_i using the posterior predictive classifier.
- (iii) Repeat (i) and (ii) for all $i = 1, \dots, n_{train}$.
- (iv) Compute the one-at-a-time training error: $\sum_{i=1}^{n_{train}} I(Y_i^0 \neq Y_i)/n_{train}$

B. Joint Training Error:

- (i) Take samples $(\boldsymbol{\beta}^{[t]}, \boldsymbol{\theta}^{[t]})$ for $t = 1, \dots, T$ and sample a corresponding $\mathbf{Z}^{[t]} \sim N(\mathbf{X}\boldsymbol{\beta}^{[t]}, \boldsymbol{\Sigma}^*(\boldsymbol{\theta}^{[t]}))$. (Note that the $\mathbf{Z}^{[t]}$ are not the same as those sampled in the MCMC algorithm.)
- (ii) For $i = 1, \dots, n_{train}$, compute $\bar{p}(\boldsymbol{\omega}) = \sum_{t=1}^T I(Z_i^{[t]} > 0)/T$ and let Y_i^0 be the predicted value of Y_i using the posterior predictive classifier.
- (iii) Compute the joint training error: $\sum_{i=1}^{n_{train}} I(Y_i^0 \neq Y_i)/n_{train}$

Both the one-at-a-time and joint training errors allow for spatial dependence among the binary predictions/classifications through the latent random variable. The joint training error allows for spatial dependence only through the spatial dependence structure of the latent variables, $\boldsymbol{\Sigma}^*(\boldsymbol{\theta})$. In contrast, the one-at-a-time training error allows for spatial dependence through $\boldsymbol{\Sigma}^*(\boldsymbol{\theta})$, but also allows for spatial dependence by conditioning on the current values of the latent random variables at nearby locations, $\mathbf{Z}_{-i}^{[t]}$.

3.2.2 Simulation Study

For this simulation study, we generate a total of 45 data sets: three data sets for each of 15 unique scenarios. Each data set has a grid size of 20×20 and the scenarios vary in terms of covariate and spatial information. To compare classification rates across different covariate values, we consider five versions of the linear component of the SGLM and SGLMM. Table 1 outlines the different linear components under consideration and the corresponding parameter values. ‘Intercept’ is a model with only an intercept and is designed to examine classification rates when only spatial information is used to determine class. ‘Simple-1’ is a linear model with an intercept and a single covariate. ‘Simple-2’ is the same model as Simple-1, but with a larger coefficient value, designed to examine classification rates where there is strong covariate information. ‘Multiple’ is also designed for this purpose, but instead of large coefficients, has more covariates. ‘Confounded’ considers the case where the covariates are spatially dependent to compare the performance of the SGLM and SGLMM classifiers in the presence of spatial confounding. For each of these five linear component models, we consider three different spatial dependence structures, thus giving 15 total scenarios.

For all cases, $\boldsymbol{\Sigma}(\boldsymbol{\theta}) \equiv \boldsymbol{\Sigma}(\rho)$, where $\boldsymbol{\Sigma}(\rho)$ is the covariance matrix defined by a conditionally autoregressive model introduced above with a second-order neighborhood structure such that locations that share a corner or an edge are taken to be neighbors. For all data sets, we fix $\rho = 0.99$ and $\gamma^2 = 1$, and vary κ . Therefore, we have three spatial dependence structures for each case: SGLMM-1 ($\kappa = 0.25$), SGLMM-2 ($\kappa = 0.5$), and SGLM ($\kappa = 1$). Finally, covariate values are assigned by drawing the values from the distributions listed in Table 1.

As is often the case with spatially-referenced data, missing observations occur in clumps of missing data, rather than scattered missing points. Therefore, for this simulation study, we consider clustered test data, where the test data were taken in randomly selected clumps. To do this, we randomly selected 25 of the locations and then randomly selected four of each location’s eight neighbors (all locations, even those on the boundary, have 8 neighbors since we have access to the data over a larger region), removing any repeats and locations outside the region. Using this approach, each data set has its own set of test locations, which make up approximately 27 percent

	<i>Intercept</i> β_0	<i>Simple-1</i> $\beta_0 + \beta_1 x_1$	<i>Simple-2</i> $\beta_0 + \beta_1 x_1$	<i>Multiple</i> $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$	<i>Confounded</i> $\beta_0 + \beta_1 x_1$
β_0	0.1	0.1	0.1	0.1	0.1
β_1	—	$-\sqrt{2}$	$-\sqrt{8}$	$-\sqrt{2}$	$-\sqrt{2}$
β_2, β_3	—	—	—	2, 2	—
x_1	—	$\sim U(-0.5, 0.5)$	$\sim U(-0.5, 0.5)$	$\sim U(-0.5, 0.5)$	$\sim \mathcal{N}(0, (I - 0.99W)^{-1})$
x_2, x_3	—	—	—	$\sim U(0, 0.5), \sim U(-0.5, 0)$	—
ρ	0.99	0.99	0.99	0.99	0.99
κ	0.25, 0.5, 1	0.25, 0.5, 1	0.25, 0.5, 1	0.25, 0.5, 1	0.25, 0.5, 1

Table 1: Parameter values and settings under which the various simulated data sets were generated.

of the locations. We use the same prior distributions for all scenarios, namely, $\beta \sim \mathcal{N}(\mathbf{0}, 10 \times \mathbf{I})$, $\rho \sim U(0, 1)$, $\kappa \sim U(0, 1)$, and for the working parameter in the data augmentation algorithms (see Appendix B.1) we use a scaled-inverse χ^2 distribution, with 3 as the scale and degrees of freedom. For each simulated data set, we fit the models using the algorithms described in Appendix B.1, run conservatively for 20,000 iterations to insure convergence. We evaluated convergence using Geweke (1992)’s convergence criteria.

We include exact error rates in Appendix C, but refer here to Figures 4 and 5 to more clearly compare the two classifiers. Figure 4 shows the misclassification error rates for all data sets, where the x-axis represents the amount of spatial dependence under which the data set was generated (κ). As expected, this figure shows that in general, the classification rules tend to perform better when there is stronger spatial dependence and stronger covariate information; the scenarios producing the smallest error rates corresponds to the data sets generated with the large coefficient value (Simple-2) and the spatially-dependent covariates (Confounded), with maximal spatial dependence ($\kappa = 1$). Furthermore, there is not much difference between the error rates for the SGLM- and SGLMM-based classifiers, regardless of which model was used to generate the data. This is more clearly seen in Figure 5, where the differences between the error rates of the SGLM and SGLMM are shown for each data set. In this case, there is not a strong pattern across the scenarios indicating when a particular classifier might be preferred over another. We take this as evidence of the robustness of the SGLM/SGLMM-based classifiers to model misspecification and find no evidence that the SGLMM is more complex than the SGLM despite its additional parameter. For one-at-a-time training error rates, in this small sample of data sets, it does appear that the SGLM fits better than the SGLMM when the true model contains stronger spatial dependence.

4 Empirical Comparison of Classification Methods

4.1 Land Cover Application

To illustrate the SGLM/SGLMM classifiers and other commonly used classification methods, we provide an empirical example using satellite-derived observations of land cover over Southeast Asia. Over the last century, Southeast Asia has experienced much deforestation, where as much as 12 percent of original forests have been lost to other land uses (Munroe et al., 2008). Researchers are interested in tracking the changing land covers to determine economic, geographic, social, and demographic factors that may contribute to deforestation. Satellite measurements provide a record

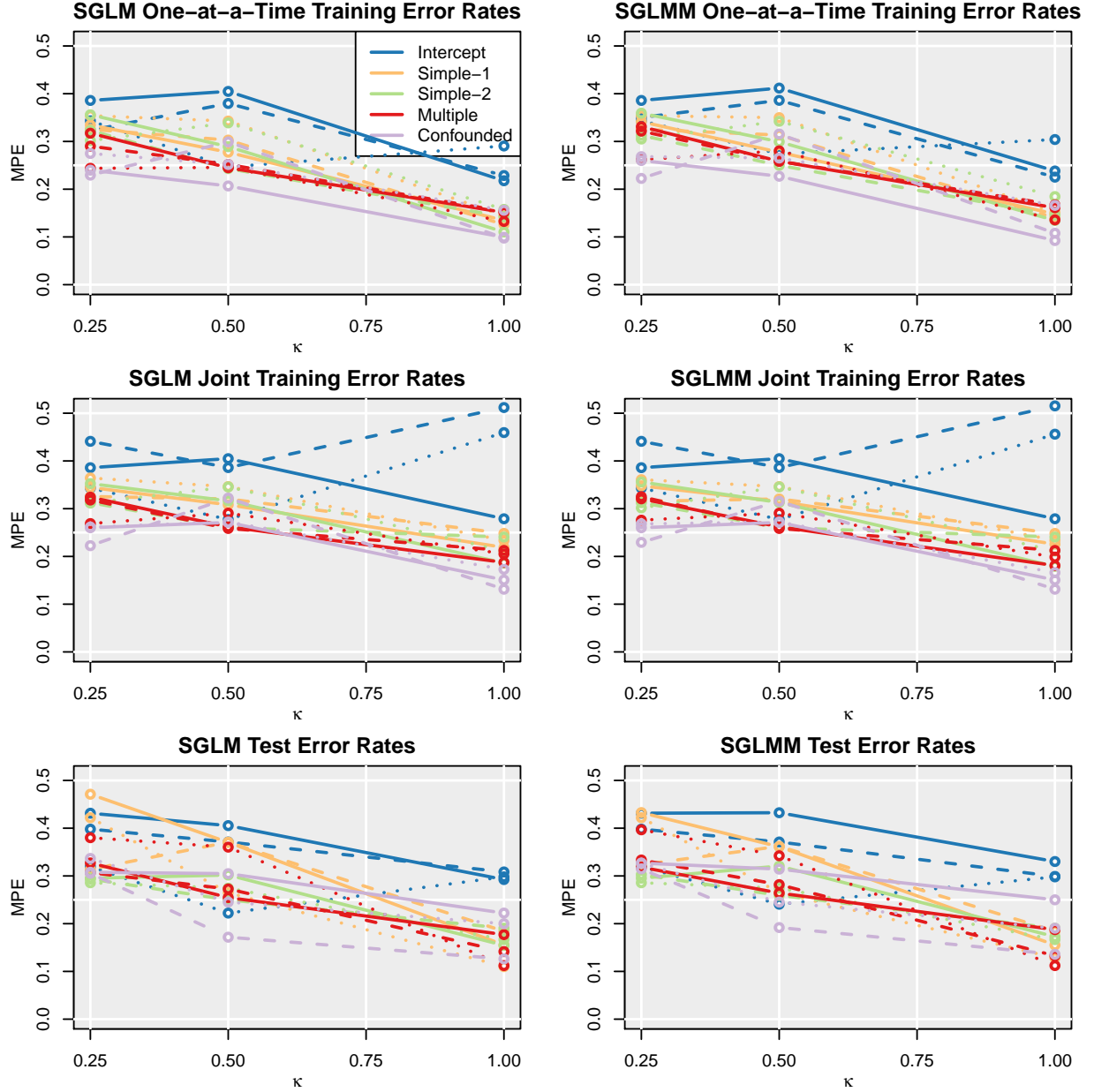


Figure 4: Classification error rates for each data set. The colors indicate the model from which the data are generated: ‘Intercept’ (green); ‘Simple-1’ (blue); ‘Simple-2’ (orange); ‘Multiple’ (red); and ‘Confounded’ (purple). The three different line types represent the three different data sets for each scenario. As a function of κ (which together with the colors completes the specification of the model that generated the data), the top row summarizes the one-at-a-time training error rates, the middle row summarizes the joint training error rates, and the bottom summarizes the test error rates. The left column contains the error rates when the SGLM is fit to the data, and the right column contains the error rates when the SGLMM is fit to the data.

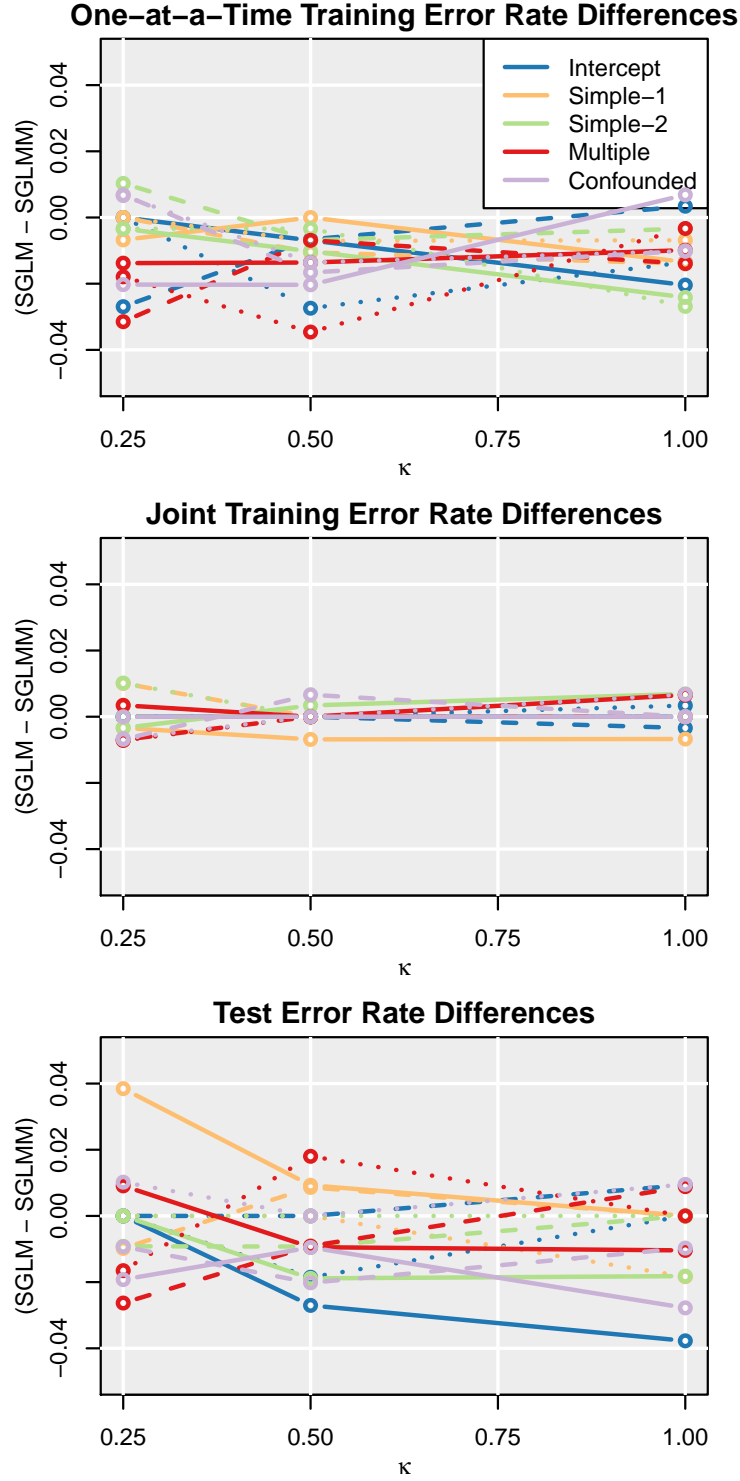


Figure 5: Differences in the error rates between the SGLM and SGLMM classifiers. Below 0 indicates the SGLM-based classifier performed better, while above 0 indicates the SGLMM-based classifier performed better.

of historic land use, but in many instances, cloud cover and other weather events may prevent the satellite from obtaining a consistent measurement for determining land cover at every location.

We used data from the National Aeronautics and Space Exploration (NASA)’s Moderate Resolution Imaging Spectroradiometer (MODIS) Land Cover Type Yearly Level 3 Global 500m (MOD12Q1 and MCD12Q1) data product for the year 2005. Our region of interest covers the region bounded by 17° to 19°N and 98° to 100°E , which covers a portion of northwestern Thailand and a small part of Myanmar. Because observations from this data are made at such a fine resolution, for computational convenience we collapsed the data into a 24×24 grid (each cell is approximately $8.8 \text{ km} \times 9.25 \text{ km}$) by selecting the most common land cover – forest or non-forest – in each grid cell.

We considered four covariates: elevation, distance to the nearest major road, distance to the coast, and distance to the nearest big city. Elevation is measured in meters and distances are Euclidean and measured in degrees. The covariates are standardized, meaning that there were no costs taken into account in calculating distance (e.g., distance calculations do not take into account the fact that it might take longer to go over mountains than go around them). To find the covariate values for the grid cells, for each covariate, we used the median value of all the observed locations within the grid cell.

With satellite data, missing observations often occur in clumps of missing data, rather than scattered points of missing data. Because of this, we randomly selected two test data sets: non-clustered and clustered. For the non-clustered test data, we randomly selected $n_{test} = 144$ locations as test data and used the remaining $n_{train} = 432$ as training data. For the clustered test data, we followed a similar method as described in Section 3.2.2. We randomly selected 36 locations and then randomly selected four of each location’s eight neighbors (all locations, even those on the boundary, have 8 neighbors since we have access to the data over a larger region), removing any repeats and locations outside the region. This resulted in $n_{test} = 159$ locations assigned to the clustered test data, and $n_{train} = 417$ assigned to the training data. Figure 6 shows images of both the non-clustered and clustered training and test data sets. We examined training and test error rates for each of these test data sets; however, we also examined test error rates for two more test data sets for each type of missing data.

4.2 Prior and Tuning Parameter Values

The Bayesian probit models require prior distributions on parameter values. For the independent probit model (i.e., $\Sigma^*(\theta)$ is the $n \times n$ identity matrix in (7)), SGLM, and SGLMM, we assign prior distribution $\beta \sim \mathcal{N}(\mathbf{0}, 10 \times \mathbf{I})$, where β are the coefficients for the centered and scaled covariates. For the working parameter in the data augmentation algorithms (see Appendix B.1), we use a scaled-inverse χ^2 distribution, with $a_\gamma = 3$ degrees of freedom and $b_\gamma = 3$ as the scale parameter. For the spatial dependence parameter in the SGLM and SGLMM, $\theta \sim \text{U}(a_\theta, b_\theta)$. We used a conditionally autoregressive (CAR) dependence structure with a second-order neighborhood structure and set $a_\theta = 0$ and $b_\theta = 1$ (since the spatial neighborhood matrix, \mathbf{W} , is row-standardized, the maximal value of ρ is 1 and we use a lower prior bound of 0 since we expect spatial dependence to be non-negative). For the SGLMM, we assign κ a $\text{Unif}(0, 1)$ prior distribution. To fit each of these models, we ran the MCMC for 120,000 iterations, at which point all chains are well-converged as

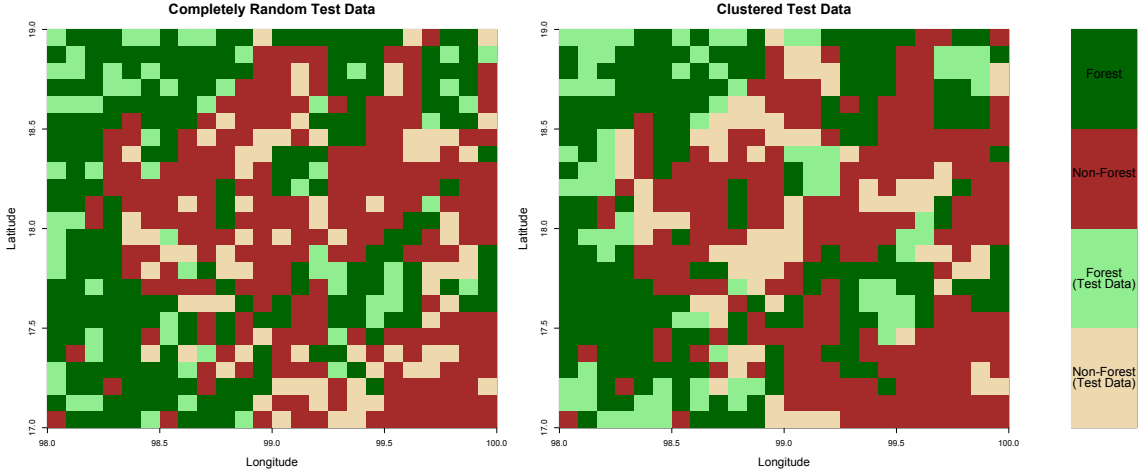


Figure 6: Non-clustered (left) and clustered (right) training (dark colors) and test (light colors) data sets. Forested locations are in green and non-forest locations are brown.

Classification Method	Tuning Parameter	<i>Non-Clustered</i>		<i>Clustered</i>	
		Optimal Value	CVE	Optimal Value	CVE
Linear SVM	λ	0.39	0.2767	0.51	0.3084
Cubic SVM	λ	0.98	0.2837	0.8	0.2964
Radial SVM	λ, u	0.515; 5.152	0.2140	0.875, 7.778	0.2410
kNN-C	k	5	0.2326	12	0.2506
kNN-G	k	5	0.1953	3	0.1880

Table 2: Optimal value of tuning parameters and the associated five-fold cross-validation errors for each classification method and data set used for the results in Table 3.

determined by examining trace plots of the parameters and Geweke (1992)’s diagnostic, and use the last 100,000 for prediction and determining error rates.

We follow the approach of Hughes and Haran (2013) for fitting the low-rank probit SGLMM (label “Hughes and Haran” below). Namely, we use the eigenvectors of the Moran operator corresponding to the largest 10 percent of the eigenvalues for the values of Ψ (see Appendix B.2). We use a $\mathcal{N}(\mathbf{0}, 100\mathbf{I})$ as the prior distribution for the coefficients of Ψ .

Support vector machines (SVM) and k-nearest neighbors (kNN) classification methods require tuning parameters, λ , u , and k (see Appendix A). We obtain optimal values of these parameters using five-fold cross-validation on each training data and assigned the value of the associated parameter to be the value with the lowest cross-validation error (CVE). Table 2 shows the chosen tuning parameter and associated CVEs for each classification method and data set.

4.3 Results

We compared error rates for the model-based classifiers discussed in Section 2.2 with additional classification methods described in Section A: linear, diagonal, and quadratic discriminant analysis

classifiers (LDA, DLDA, and QDA, respectively); linear, cubic, and radial support vector machines (SVM); k-nearest neighbor (kNN) classifiers where neighbors are defined either in covariate space (kNN-C; the traditional approach) or geographic space (kNN-G); and four spatial classifiers from Switzer (1980) (“Switzer”), Mardia (1984) (“Mardia”), Šaltytė Benth and Dučinskas (2005) (Spatial LDA), and Press (1996) (“Press”). We examine in-depth the training and test errors for a single test data set in Table 3. We also examine test error rates for more test data sets in Table 4. We considered two scenarios of missing data: non-clustered and clustered training/test data sets (see Figure 6).

We first compare training and test error rates for the first sets of test data, shown in Table 3. The training and test error rates tend to be similar for each classification method. Furthermore, the non-clustered and clustered data sets have similar error rates. Most of the error rates from non-spatial methods fall around 30 percent; however, the classification methods which make use of neighboring classes (i.e., the SGLM, SGLMM, and kNN-G classifiers) have much lower error rates.

As in Section 3.2.2, the SGLM and SGLMM error rates are similar, with the SGLM doing slightly better in this case. While not included here, we note that the reported error rates were similar when we used a geostatistical-type exponential covariance structure rather than a CAR model. Therefore, we do not believe the differences in error rates are a consequence of the particular form of the spatial dependence structure and for computational convenience we prefer the CAR model.

As expected, the SGLM and SGLMM joint training error rates are not as small as the one-at-a-time training error rates which has the added benefit of using of the observed classes of neighboring locations and not just allowing for spatial dependence in the residual of the latent variable. The one-at-a-time training error rates are also more similar to the test error rates, implying that the one-at-a-time training error rates would be more useful in a practical classification setting.

For the clustered test data, the SGLM and SGLMM-based classifiers have the lowest error rates. kNN-G also uses neighboring classes and has small error rates relative to the other competing methods. Although this classifier has smaller rates for the training data, it does not have the smallest error rates across the board. Furthermore, the error rates for this classifier are not fixed because of the random classification when there are ties. For example, repeating this classification method several times for the non-clustered data, the test error rates for kNN-G ranged from 12 percent up to 21 percent. Although including only the classes of neighboring locations improves the error rates over those methods which do not make use of neighboring classes, also including covariates in the classifier – as in the SGLM or SGLMM – provides as low and more consistent error rates.

SVM classification methods offer more flexibility in the relationship between the inputs and the probability of the classes than a GLM. The radial SVM classifier does quite well, and in fact has the smallest training error rates. However, the test error rates are not as small as those for the spatial methods, suggesting that allowing for this more complicated structure among the inputs is not sufficient to produce better predictions when classifying out-of-sample data in space.

The other spatial classifiers (“Switzer”, “Mardia”, Spatial LDA, and “Press”) make use of spatial structure in the covariates/inputs. Although the “Switzer” classifier is competitive with LDA, QDA, and SVMs, the other spatial classifiers – including the Bayesian classifier (“Press”) – perform rather

Classification Method	<i>Non-clustered Data</i> Training	Test	<i>Clustered Data</i> Training	Test
Model-based Classifiers				
<i>SGLM</i>				
Posterior Mean	0.1551	0.1667	0.1775	0.1447
Posterior Predictive	0.1551; 0.2731	0.1667	0.1775; 0.2998	0.1447
<i>SGLMM</i>				
Posterior Mean	0.1644	0.1736	0.1823	0.1447
Posterior Predictive	0.1829; 0.2894	0.1667	0.1823; 0.3046	0.1447
Hughes and Haran	0.1296	0.2222	0.1367	0.1696
<i>Probit - Bayesian</i>				
Posterior Mean	0.2824	0.2986	0.2854	0.2767
Posterior Predictive	0.2824	0.2986	0.2854	0.2767
<i>GLM - Maximum Likelihood</i>				
Logistic	0.2824	0.3056	0.2854	0.2767
Probit	0.2824	0.2986	0.2854	0.2767
Additional Classifiers				
<i>Discriminant Analysis</i>				
LDA	0.2778	0.2917	0.2878	0.2704
DLDA	0.3611	0.3889	0.3549	0.3522
QDA	0.2593	0.2986	0.2638	0.2767
<i>SVM</i>				
Linear SVM	0.2847	0.3264	0.2926	0.2956
Cubic SVM	0.2685	0.2847	0.2806	0.3396
Radial SVM	0.0949	0.2639	0.0530	0.1950
<i>k-Nearest Neighbors</i>				
kNN-C	0.1597	0.2639	0.2062	0.2579
kNN-G	0.1204	0.1458	0.1055	0.2075
<i>Spatial Classifiers</i>				
Switzer	0.2616	0.2639	0.2566	0.2579
Mardia	0.4074	0.4375	0.5348	0.5094
Spatial LDA	0.3634	0.3750	0.3741	0.4717
Press	0.3912	0.3611	0.3933	0.4214

Table 3: Training and test errors for the SE Asia land cover data obtained using various classification methods. The posterior predictive errors for the training data list two errors: one-at-a-time (left) and joint (right).

poorly in comparison to the other classification methods. This is perhaps due to the fact that the spatial structure of the covariates differs from that of the process of interest (forest/not forest). In this case, we want to take advantage of the covariate information at the focal location, but not at the neighboring locations. Comparing the error rates of all these classification methods suggests

Classification Method	<i>Non-clustered Data</i>		<i>Clustered Data</i>	
	Test Set 1	Test Set 2	Test Set 3	Test Set 4
Model-based Classifiers				
<i>SGLM</i>				
Posterior Mean	0.1806	0.1806	0.1688	0.2318
Posterior Predictive	0.1806	0.1806	0.1688	0.2318
<i>SGLMM</i>				
Posterior Mean	0.1806	0.1944	0.1688	0.2252
Posterior Predictive	0.1806	0.1944	0.1623	0.2252
Hughes and Haran	0.1875	0.1875	0.1688	0.2715
<i>Probit - Bayesian</i>				
Posterior Mean	0.2986	0.3056	0.2143	0.3245
Posterior Predictive	0.2917	0.3056	0.2013	0.3245
<i>GLM - Maximum Likelihood</i>				
Logistic	0.2986	0.3125	0.2078	0.3311
Probit	0.2986	0.3056	0.2143	0.3245
Additional Classifiers				
<i>Discriminant Analysis</i>				
LDA	0.2917	0.3264	0.2143	0.3179
DLDA	0.3681	0.4167	0.3377	0.5033
QDA	0.2847	0.3194	0.2143	0.3510
<i>SVM</i>				
Linear SVM	0.2986	0.3264	0.2143	0.3245
Cubic SVM	0.2917	0.3333	0.2727	0.3775
Radial SVM	0.2847	0.3056	0.2143	0.3510
<i>k-Nearest Neighbors</i>				
kNN-C	0.2708	0.2917	0.2792	0.3444
kNN-G	0.2361	0.2847	0.3117	0.2715
<i>Spatial Classifiers</i>				
Switzer	0.2639	0.3056	0.1753	0.2715
Mardia	0.4514	0.4583	0.4416	0.4437
Spatial LDA	0.4514	0.4583	0.4416	0.4437
Press	0.5556	0.5903	0.5584	0.4636

Table 4: Test errors under various classification methods for multiple test data sets for the SE Asia land cover data. The bolded test errors are the smallest test errors across all classification methods for that particular test set.

that including inputs associated with the focal location as well as neighboring observations, as in the SGLM and SGLMM classifiers, leads to better classification methods.

Table 4 shows similar test error rates to those found in Table 3. However, the SGLM- and SGLMM-based classifiers perform the best across the board. We also note that the Hughes and Haran

approach which accounts for spatial confounding between the covariates and the spatial random effect fits almost as well as the full SGLM- and SGLMM-based classifiers, but not quite. We assume this is because of the dimension reduction which smoothes out some of the spatial dependence, which may be important in making use of neighboring pixel classes.

In terms of computation time, the SGLM has a large computational advantage over the SGLMM, as it has one less parameter and, for the CAR dependence structure, it does not require inverting the $n \times n$ covariance matrix. For the 120,000 iterations run for this application, the SGLM took 17.74 CPU hours running on a single core of an Intel Xeon CPU E7-8837 2.66GHz, while the SGLMM took 50.50 CPU hours using up to 18 cores where possible. Fitting these classifiers are much more computationally intensive than the others described in this paper, where the non-Bayesian methods are nearly instantaneous. While perhaps some would consider this large difference in computational time, we note that it may be worth it. In our example from Table 3, the SGLM classified 6 percent more of the locations correctly than the next best classifier (kNN-G). Furthermore, since the SGLM- and SGLMM-based classifiers are based on an underlying statistical model, we can obtain measures of uncertainty that the algorithmic-based classification methods cannot. Lastly, we note that we ran the MCMC algorithms longer than is likely needed in order to be conservative. If fitting time was a major concern, a less conservative strategy might be warranted, such as the dimension reduced “Hughes and Haran” SGLMM, which required only 1.4 CPU hours.

5 Discussion

In this paper, we discuss Bayesian classifiers for spatial data based on the Bayesian probit SGLM and SGLMM. Although these models are similar, we attempt to provide some insight into their differences. Through two illustrative examples and a simulation study, we find that classification rules based on these models are fairly robust to model misspecification. Another contribution of this paper is a comprehensive review and empirical comparison of alternative classifiers in the literature. While the SGLM/SGLMM-based classifiers outperform others in the literature for our land cover application, we caution against drawing general conclusions from these findings. For other applications, the performance of the classifiers may differ. We hope that our overview of spatial classification methods will be of use to researchers in selecting candidate methods for other applications.

We focused on building spatial classifiers using a probit SGLM and SGLMM, but other spatial statistical models for discrete data may also suggest classification rules that include neighboring location classes and covariates as inputs. For example, Hoeting et al. (2000) developed a spatial classifier based on the autologistic model. A comparison of alternative spatial model-based classifiers is left to future work. In addition, we note that many of the other classification methods cited in this work allow for classification of more than two categories. Albert and Chib (1993) extend their latent variable probit model to ordinal and categorical response variables. Using this framework, the SGLM and SGLMM have been extended to model these types of response variables in a spatial setting (see, for example, Berrett, 2010; Schliep and Hoeting, 2013). Classifiers based on these model extensions could be derived in a similar fashion and compared to the other classification methods. Also, we note that throughout this paper, we assume that the unobserved locations are missing completely at random (MCAR), meaning that the locations of the missing data are unrelated to other variables. Of course, this may not be the case in practice, and future

work could investigate the utility of these classifiers under other missing data scenarios.

Finally, we note that in the empirical analysis in Section 4 we used a collapsed version of the original data due to computational considerations. The original data consisted of 54,776 observations. Although fitting any of the classification methods provided here to the full data set will be computationally challenging, we note that low-rank methods such as those discussed within the paper (Reich et al., 2006; Hughes and Haran, 2013; Hanks et al., 2015) can help ease the computational burden of fitting the SGLMM.

Acknowledgements

The authors gratefully acknowledge support for this work from NASA’s Land-Cover/Land-Use Change Program (NNG06GD31G), as well as from the National Science Foundation (DMS-0934595 and DMS-1209161). They also thank Dr. Ningchuan Xiao, Department of Geography at The Ohio State University, for his assistance working with the land cover data, and the three anonymous referees for their helpful comments and suggestions for improving this paper.

References

- Albert, J. H. and Chib, S. (1993). “Bayesian analysis of binary and polychotomous response data.” *Journal of the American Statistical Association*, 88, 669–679.
- Banerjee, S., Carlin, B., and Gelfand, A. (2004). *Hierarchical Modeling and Analysis for Spatial Data*. Boca Raton, FL: Chapman & Hall/CRC.
- Batsidis, A. and Zografos, K. (2011). “Errors of misclassification in discrimination of dimensional coherent elliptic random field observations.” *Statistical Neerlandica*, 65, 446–461.
- Berrett, C. (2010). *Bayesian probit regression models for spatially-dependent categorical data*. Columbus, Ohio: Ohio State University. <http://rave.ohiolink.edu/etdc/view.cgi?acc%5Fnum=osu1285076512>.
- Berrett, C. and Calder, C. A. (2012). “Data augmentation strategies for the Bayesian spatial probit regression model.” *Computational Statistics and Data Analysis*, 56, 478–490.
- Cortes, C. and Vapnik, V. (1995). “Support-vector network.” *Machine Learning*, 20, 273–297.
- De Oliveira, V. (2000). “Bayesian prediction of clipped Gaussian random fields.” *Computational Statistics and Data Analysis*, 34, 299–314.
- Diggle, P. J., Tawn, J. A., and Moyeed, R. A. (1998). “Model-based geostatistics.” *Applied Statistician*, 47, 299–350.
- Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., and Weingessel, A. (2010). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.5-24.
- Geweke, J. (1992). “Evaluating the accuracy of sampling-based approaches to calculating posterior moments.” In *Bayesian Statistics 4*, eds. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith. Oxford, UK: Clarendon Press.

- Hanks, E. M., Schliep, E. M., Hooten, M. B., and Hoeting, J. A. (2015). “Restricted spatial regression in practice: Geostatistical models, confounding, and robustness under model misspecification.” *Environmetrics*, 26, 243–254.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer.
- Higgs, M. D. and Hoeting, J. A. (2010). “A clipped latent-variable model for spatially correlated ordered categorical data.” *Computational Statistics and Data Analysis*, 54, 1999–2011.
- Hoeting, J. A., Leecaster, M., and Bowden, D. (2000). “An Improved Model for Spatially Correlated Binary Responses.” *Journal of Agricultural, Biological, and Environmental Statistics*, 5, 102–114.
- Hughes, J. and Haran, M. (2013). “Dimension reduction and alleviation of confounding for spatial generalized linear mixed models.” *Journal of the Royal Statistical Society, Series B*, 75, 139–159.
- Imai, K. and van Dyk, D. A. (2005). “A Bayesian analysis of the multinomial probit model using marginal data augmentation.” *Journal of Econometrics*, 124, 311–334.
- Klein, R. and Press, S. J. (1992). “Adaptive Bayesian classification of spatial data.” *Journal of the American Statistical Association*, 87, 844–851.
- Mardia, K. V. (1984). “Spatial discrimination and classification maps.” *Communications in Statistics, Theory and Methods*, 13, 2181–2197.
- McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. 2nd ed. London: Chapman & Hall/CRC.
- Munroe, D. K., Wolfenbarger, S. R., Calder, C. A., Shi, T., Xiao, N., Lamb, C. Q., and Li, D. (2008). “The relationships between biomass burning, land-cover/-use change, and the distribution of carbonaceous aerosols in mainland Southeast Asia: a review and synthesis.” *Journal of Land Use Science*, 3, 161–183.
- Paciorek, C. J. (2007). “Computational techniques for spatial logistic regression with large data sets.” *Computational Statistics and Data Analysis*, 51, 3631–3653.
- Press, S. J. (1996). “The directional neighborhoods approach to contextual classification of images from noisy data.” *Journal of the American Statistical Association*, 91, 1091–1100.
- Reich, B. J., Hodges, J. S., and Zadnik, V. (2006). “Effects of residual smoothing on the posterior of the fixed effects in disease-mapping models.” *Biometrics*, 62, 4, 1197–1206.
- Schliep, E. M. and Hoeting, J. A. (2013). “Multivariate multilevel latent Gaussian process model to evaluate wetland condition.” *Journal of Agricultural, Biological, and Environmental Statistics*, 18, 492–513.
- Switzer, P. (1980). “Extension of discriminant analysis for statistical classification of remotely sensed satellite imagery.” *Journal of the International Association for Mathematical Geology*, 12, 86–95.
- Šaltytė Benth, J. and Dučinskas, K. (2005). “Linear discriminant analysis of multivariate spatial-temporal regressions.” *Scandinavian Journal of Statistics*, 32, 281–294.

APPENDICES

A Other Classification Methods

In this appendix, we provide decision functions for the other classification methods discussed in Section 2. For these alternative methods, we assume that the \mathbf{x} s only include the inputs, and thus do not include a term to allow for an intercept as in the model-based (GLM/GLMM) classification methods. Unless otherwise noted, the descriptions of the classification methods are based on Hastie et al. (2001).

A.1 Model-based Non-spatial Classification

In this section we consider the corresponding non-spatial versions to the model-based spatial classification methods derived in Section 2. For non-spatially dependent binary data, the underlying model is defined by equations (1)–(3), where $\nu_i \equiv 0$ for all i . In this case, $\boldsymbol{\omega} = (\mathbf{x}^0, \boldsymbol{\beta})$,

$$p_1(\boldsymbol{\omega}) \equiv p_1(\mathbf{x}^0, \boldsymbol{\beta}) = g^{-1}(\mathbf{x}^{0'}\boldsymbol{\beta}),$$

and $p_0(\boldsymbol{\omega}) = 1 - p_1(\boldsymbol{\omega})$. It follows that the decision function is

$$\delta_{GLM}(\boldsymbol{\omega}) \equiv \delta_{GLM}(\mathbf{x}^0, \boldsymbol{\beta}) = \frac{p_1(\mathbf{x}^0, \boldsymbol{\beta})}{p_0(\mathbf{x}^0, \boldsymbol{\beta})} = \frac{g^{-1}(\mathbf{x}^{0'}\boldsymbol{\beta})}{1 - g^{-1}(\mathbf{x}^{0'}\boldsymbol{\beta})}. \quad (16)$$

For the logit link function,

$$\delta_{GLM-L}(\boldsymbol{\omega}) = \delta_{GLM-L}(\mathbf{x}^0, \boldsymbol{\beta}) = \exp\{\mathbf{x}^{0'}\boldsymbol{\beta}\}, \quad (17)$$

and for the probit link function,

$$\delta_{GLM-P}(\boldsymbol{\omega}) = \delta_{GLM-P}(\mathbf{x}^0, \boldsymbol{\beta}) = \frac{\Phi(\mathbf{x}^{0'}\boldsymbol{\beta})}{1 - \Phi(\mathbf{x}^{0'}\boldsymbol{\beta})}. \quad (18)$$

For both decision functions, we can obtain maximum likelihood, posterior mean, and posterior predictive classifiers. In practice, we evaluate the integral in (11) via Monte Carlo integration: we approximate $\bar{p}_1(\boldsymbol{\omega})$ by drawing a realization of $Y^{0[t]} \sim \text{Bernoulli}(p_1(\mathbf{x}^0, \boldsymbol{\beta}^{[t]}))$ for $t = 1, \dots, T$ and setting $\bar{p}_1(\boldsymbol{\omega}) = \sum_{t=1}^T I(Y^{0[t]} = 1)/T$, where $\boldsymbol{\beta}^{[t]}$ are the draws from the posterior distribution of $\boldsymbol{\beta}$, and T is the number of draws from the posterior distribution.

A.2 Discriminant Analysis

In discriminant analysis, rather than considering the explanatory variables \mathbf{x} as fixed as they are in regression analyses, the \mathbf{x} s are viewed as random variables, with class-specific density functions $f_j(\mathbf{x})$ corresponding to each class C_j . The classes also have prior probabilities π_j , such that $\pi_0 + \pi_1 = 1$. To determine the probability that a set of inputs will fall into class j , we employ Bayes' theorem which implies that

$$P(C_j|\mathbf{x}) = \frac{f_j(\mathbf{x})\pi_j}{f_1(\mathbf{x})\pi_1 + f_0(\mathbf{x})\pi_0}. \quad (19)$$

As with the model-based classification methods, the Bayes' classifier is to classify an observation to class C_1 when $P(C_1|\mathbf{x}) > P(C_0|\mathbf{x})$ and to C_0 otherwise.

We first describe discriminant analysis in its general form, allowing an arbitrary form for the $f_j(\mathbf{x})$ s and the decision function, and then discuss three special cases that we use in our analysis. Let

$$\mathbf{X}_j = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{n_j} \end{bmatrix}$$

where $\{\mathbf{x}_1, \dots, \mathbf{x}_{n_j}\} = \{\mathbf{x}_i : y_i = j\}$ and \mathbf{X}_j is an $n_j \ell \times 1$ vector of inputs corresponding to observations in class C_j . To classify Y^0 , for each class we define

$$\mathbf{X}_j^0 = \begin{bmatrix} \mathbf{x}^0 \\ \mathbf{X}_j \end{bmatrix},$$

where \mathbf{X}_j^0 is an $(n_j + 1)\ell \times 1$ vector and \mathbf{x}^0 is an $\ell \times 1$ vector of inputs associated with Y^0 . Our goal is to determine a decision boundary for classifying Y^0 .

In discriminant analysis, $f_j(\cdot)$ is typically the multivariate normal density function,

$$f_j(\mathbf{X}_j^0) = \frac{1}{(2\pi)^{(n_j+1)\ell/2} |\boldsymbol{\Sigma}_j^X|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{X}_j^0 - \boldsymbol{\mu}_j^X)' (\boldsymbol{\Sigma}_j^X)^{-1} (\mathbf{X}_j^0 - \boldsymbol{\mu}_j^X) \right\},$$

where $\boldsymbol{\mu}_j^X$ is the $(n_j + 1)\ell \times 1$ class-specific mean vector and $\boldsymbol{\Sigma}_j^X$ is the $(n_j + 1)\ell \times (n_j + 1)\ell$ class-specific covariance matrix. It follows that

$$f_j(\mathbf{x}^0 | \mathbf{x}_1, \dots, \mathbf{x}_{n_j}) = \frac{1}{(2\pi)^{k/2} |\boldsymbol{\Sigma}_j^{x^0}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}^0 - \boldsymbol{\mu}_j^{x^0})' (\boldsymbol{\Sigma}_j^{x^0})^{-1} (\mathbf{x}^0 - \boldsymbol{\mu}_j^{x^0}) \right\} \quad (20)$$

where

$$\begin{aligned} \boldsymbol{\mu}_j^{x^0} &= \boldsymbol{\mu}_{j(\{1:\ell\})} + \boldsymbol{\Sigma}_{j(\{1:\ell\}, -\{1:\ell\})}^X \left(\boldsymbol{\Sigma}_{j(-\{1:\ell\}, -\{1:\ell\})}^X \right)^{-1} (\mathbf{X}_j - \boldsymbol{\mu}_{j(-\{1:\ell\})}) \\ \boldsymbol{\Sigma}_j^{x^0} &= \boldsymbol{\Sigma}_{j(\{1:\ell\}, \{1:\ell\})}^X - \boldsymbol{\Sigma}_{j(\{1:\ell\}, -\{1:\ell\})}^X \left(\boldsymbol{\Sigma}_{j(-\{1:\ell\}, -\{1:\ell\})}^X \right)^{-1} \boldsymbol{\Sigma}_{j(-\{1:\ell\}, \{1:\ell\})}^X. \end{aligned}$$

We use the subscript notation $(\{1:\ell\})$ to indicate the first ℓ elements of the corresponding matrix or vector (i.e., those indices corresponding to \mathbf{x}^0) and $(-\{1:\ell\})$ indicates the matrix or vector without the first ℓ elements (i.e., the remaining indices corresponding to \mathbf{X}_j).

Considering the log-odds, it follows from (19) and (20) that

$$\begin{aligned} \log \left(\frac{P(C_1|\mathbf{x}^0)}{P(C_0|\mathbf{x}^0)} \right) &= \underbrace{\log \frac{\pi_1}{\pi_0} + \frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_1^{x^0}|}{|\boldsymbol{\Sigma}_0^{x^0}|} - \frac{1}{2} \boldsymbol{\mu}_1^{x^0'} (\boldsymbol{\Sigma}_1^{x^0})^{-1} \boldsymbol{\mu}_1^{x^0} + \frac{1}{2} \boldsymbol{\mu}_0^{x^0'} (\boldsymbol{\Sigma}_0^{x^0})^{-1} \boldsymbol{\mu}_0^{x^0}}_{\equiv \alpha_0} \\ &\quad + \underbrace{\mathbf{x}^{0'} \left((\boldsymbol{\Sigma}_1^{x^0})^{-1} \boldsymbol{\mu}_1 - (\boldsymbol{\Sigma}_0^{x^0})^{-1} \boldsymbol{\mu}_0^{x^0} \right)}_{\equiv \alpha_1} - \underbrace{\mathbf{x}^{0'} \frac{1}{2} \left((\boldsymbol{\Sigma}_1^{x^0})^{-1} - (\boldsymbol{\Sigma}_0^{x^0})^{-1} \right) \mathbf{x}^0}_{\equiv \alpha_2}. \end{aligned} \quad (21)$$

Here, α_0 is a scalar, α_1 is an $\ell \times 1$ vector, and α_2 is an $\ell \times \ell$ matrix, which we define for notational convenience. A *discriminant-analysis decision function* corresponding to the classification rule defined in equation (10) is

$$\delta_{DA}(\omega) \equiv \delta_{DA}(\mathbf{x}^0, \boldsymbol{\mu}_0^{x^0}, \boldsymbol{\mu}_1^{x^0}, \boldsymbol{\Sigma}_0^{x^0}, \boldsymbol{\Sigma}_1^{x^0}) = \exp\{\alpha_0 + \mathbf{x}^{0'} \alpha_1 - \mathbf{x}^{0'} \alpha_2 \mathbf{x}^0\}. \quad (22)$$

We now consider special cases of (22). Each of these special cases assumes that the mean of \mathbf{x}_i is equal across all observations, so that $\boldsymbol{\mu}_j^X = [\mathbf{1}_{n_j+1} \otimes \boldsymbol{\mu}_j]$ where $\mathbf{1}_{n_j+1}$ is an $(n_j + 1) \times 1$ vector of ones and $\boldsymbol{\mu}_j$ is an $\ell \times 1$ class specific mean vector. The difference between each of these special cases is in the specification of the covariance matrix $\boldsymbol{\Sigma}_j^X$. We describe three popular discriminant analysis methods (linear discriminant analysis, diagonal linear discriminant analysis, and quadratic discriminant analysis) all of which assume that the \mathbf{x}_i are independent.

Assuming the \mathbf{x}_i are independent results in the following form for the covariance of \mathbf{X}_j^0 :

$$\text{var}(\mathbf{X}_j^0) = \boldsymbol{\Sigma}_j^X = (\mathbf{I}_{n_j+1} \otimes \boldsymbol{\Lambda}_j), \quad (23)$$

where \mathbf{I}_{n_j+1} is an $(n_j + 1) \times (n_j + 1)$ identity matrix and $\boldsymbol{\Lambda}_j$ is a class-specific covariance matrix for the ℓ components of \mathbf{x}_i . Under this assumption, \mathbf{x}^0 is independent of $\mathbf{x}_1, \dots, \mathbf{x}_{n_j}$, so

$$f_j(\mathbf{x}^0 | \mathbf{x}_1, \dots, \mathbf{x}_{n_j}) = f_j(\mathbf{x}^0) = \frac{1}{(2\pi)^{k/2} |\boldsymbol{\Lambda}_j|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}^0 - \boldsymbol{\mu}_j)' \boldsymbol{\Lambda}_j^{-1} (\mathbf{x}^0 - \boldsymbol{\mu}_j)\right\}.$$

Assuming a constant variance across classes (i.e., $\boldsymbol{\Lambda}_j = \boldsymbol{\Lambda}$ for $j = 0, 1$) results in linear discriminant analysis (LDA) because the decision boundary is linear in the \mathbf{x} 's. The log odds in equation (21) can be written in this case as

$$\log \left(\frac{P(C_1 | \mathbf{x}^0)}{P(C_0 | \mathbf{x}^0)} \right) = \underbrace{\log \frac{\pi_1}{\pi_0} - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)' \boldsymbol{\Lambda}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)}_{\alpha_0^{LDA}} + \mathbf{x}^{0'} \underbrace{\boldsymbol{\Lambda}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)}_{\boldsymbol{\alpha}_1^{LDA}},$$

where α_0^{LDA} is a scalar and $\boldsymbol{\alpha}_1^{LDA}$ is an $\ell \times 1$ vector, defined for notational convenience. Therefore, the decision function for LDA is

$$\delta_{LDA}(\omega) \equiv \delta_{LDA}(\mathbf{x}^0, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Lambda}) = \exp\{\alpha_0^{LDA} + \mathbf{x}^{0'} \boldsymbol{\alpha}_1^{LDA}\}. \quad (24)$$

Note that this decision function is effectively equivalent to the one based on the logistic regression model in (17), however, in logistic regression, we assume the \mathbf{x} 's are fixed and thus make no distributional assumptions on \mathbf{x} as in discriminant analysis.

In practice, the parameters $\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Lambda}$ (and thus α_0^{LDA} and $\boldsymbol{\alpha}_1^{LDA}$) are unknown but can be estimated using maximum likelihood:

- $\hat{\pi}_j = n_j/n$,
- $\hat{\boldsymbol{\mu}}_j = \sum_{i: y_i=j} \mathbf{x}_i / n_j$,

- $\hat{\Lambda} = \sum_{j \in \{0,1\}} \sum_{i:y_i=j} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j)' / (n - 2).$

Diagonal linear discriminant analysis (DLDA) additionally assumes independence between the k inputs so that $\text{var}(\mathbf{x}_i) = \Lambda$ is a diagonal matrix. The *DLDA-based decision function* is the same as (24), but using a diagonal matrix Λ . The m^{th} diagonal element of Λ is estimated by $\hat{\Lambda}_{(m,m)} = \sum_{j \in \{0,1\}} \sum_{i:y_i=j} (x_{im} - \hat{\mu}_{jm})^2 / (n - 2)$ where x_{im} and $\hat{\mu}_{jm}$ are the m^{th} elements of \mathbf{x}_i and $\boldsymbol{\mu}_j$, respectively.

In LDA, we assume a constant covariance for \mathbf{x}_i among the classes (i.e., $\Lambda_j = \Lambda$ for $j = 0, 1$). On the other hand, quadratic discriminant analysis (QDA) allows for each class to have its own covariance. The log odds now contains a quadratic term in the \mathbf{x} 's:

$$\begin{aligned} \log \left(\frac{P(C_1|\mathbf{x}^0)}{P(C_0|\mathbf{x}^0)} \right) = & \underbrace{\log \frac{\pi_1}{\pi_0} + \frac{1}{2} \log \frac{|\Lambda_0|}{|\Lambda_1|} - \frac{1}{2} \boldsymbol{\mu}_1' \Lambda_1^{-1} \boldsymbol{\mu}_1 + \frac{1}{2} \boldsymbol{\mu}_0' \Lambda_0^{-1} \boldsymbol{\mu}_0}_{\alpha_0^{QDA}} \\ & + \mathbf{x}^{0'} \underbrace{(\Lambda_1^{-1} \boldsymbol{\mu}_1 - \Lambda_0^{-1} \boldsymbol{\mu}_0)}_{\boldsymbol{\alpha}_1^{QDA}} - \mathbf{x}^{0'} \underbrace{\frac{1}{2} (\Lambda_1^{-1} - \Lambda_0^{-1})}_{\boldsymbol{\alpha}_2^{QDA}} \mathbf{x}^0. \end{aligned}$$

Here, α_0^{QDA} is a scalar, $\boldsymbol{\alpha}_1^{QDA}$ is an $\ell \times 1$ vector, and $\boldsymbol{\alpha}_2^{QDA}$ is an $\ell \times \ell$ matrix, which are defined for notational convenience. This results in the *QDA-based decision function*:

$$\delta_{QDA}(\boldsymbol{\omega}) \equiv \delta_{QDA}(\mathbf{x}^0, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \Lambda_0, \Lambda_1) = \exp\{\alpha_0^{QDA} + \mathbf{x}^{0'} \boldsymbol{\alpha}_1^{QDA} + \mathbf{x}^{0'} \boldsymbol{\alpha}_2^{QDA} \mathbf{x}^0\},$$

where we estimate the parameters by taking

- $\hat{\pi}_j = n_j / n,$
- $\hat{\boldsymbol{\mu}}_j = \sum_{i:y_i=j} \mathbf{x}_i / n_j,$
- $\hat{\boldsymbol{\Sigma}}_j = \sum_{i:y_i=j} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j)' / (n_j - 1).$

A.3 Support Vector Machines

First introduced by Cortes and Vapnik (1995), the goal of support vector machines (SVM) is to determine a hyperplane in covariate space separating the classes in such a way that the margin between the two classes is maximized. The margin is the minimum distance between the the inputs \mathbf{x}_i of the two classes in the direction perpendicular to the hyperplane. The resulting function determining this hyperplane is the decision function.

In SVM, the classes are labeled as either 1 or -1 (instead of 1 or 0, as before). To accommodate this convention, we redefine observations $y_i^* = 2y_i - 1$, for $i = 1, \dots, n$, so that $y_i^* \in \{-1, 1\}$.

Consider the decision function

$$\delta_{SVM}(\boldsymbol{\omega}) \equiv \delta_{SVM}(\mathbf{x}, \boldsymbol{\beta}, \beta_0) = \exp\{\mathbf{x}' \boldsymbol{\beta} + \beta_0\}. \quad (25)$$

Given observations $\{y_i^*, \mathbf{x}_i\}$ for $i = 1, \dots, n$, maximizing the margin between the two classes and the hyperplane is equivalent to minimizing $\|\boldsymbol{\beta}\|$ subject to $y_i^* (\mathbf{x}_i' \boldsymbol{\beta} + \beta_0) \geq 1$ for all $i = 1, \dots, n$.

This problem can be represented as the following Lagrange optimization function

$$\max_{\boldsymbol{\zeta}} L = \max_{\boldsymbol{\zeta}} \left(\sum_{i=1}^n \zeta_i - \frac{1}{2} \sum_{i=1}^n \sum_{i^*=1}^n \zeta_i \zeta_{i^*} y_i^* y_{i^*}^* \mathbf{x}_i' \mathbf{x}_{i^*} \right), \quad (26)$$

where $\boldsymbol{\zeta} = (\zeta_1, \dots, \zeta_n)'$ are the Lagrangian multipliers, which are subject to the constraints that $\sum_{i=1}^n \zeta_i y_i^* = 0$ and $\zeta_i \geq 0$ for all i . The \mathbf{x}_i where $\zeta_i > 0$ are the support vectors and are the only vectors which influence the position of the hyperplane. We can write the relationship between $\boldsymbol{\zeta}$ and $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta} = \sum_{i=1}^n \zeta_i y_i^* \mathbf{x}_i$$

and the relationship between $\boldsymbol{\zeta}$ and β_0 as

$$\beta_0 = \frac{1}{n_{sv}} \sum_{i: \zeta_i > 0} (\boldsymbol{\beta} \mathbf{x}_i - y_i^*)$$

where n_{sv} is the number of support vectors.

While the above hyperplane is linear, SVM can be extended to create nonlinear boundaries between the classes. This extension can be achieved by transforming the inputs into a space where they can be separated linearly, and again find the separating hyperplane in this transformed covariate space. We can use the Lagrange optimization function in (26) with the transformed inputs $h(\mathbf{x}_i, \mathbf{x}_j)$:

$$\max L = \max \left(\sum_{i=1}^n \zeta_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \zeta_i \zeta_j y_i^* y_j^* h(\mathbf{x}_i, \mathbf{x}_j) \right) \quad (27)$$

subject to $\sum_{i=1}^n \zeta_i y_i^* = 0$ and $0 \leq \zeta_i \leq \lambda$ for all i , where λ is a tuning parameter allowing for crossover among the two classes and $h(\cdot, \cdot)$ is a symmetric positive (semi-) definite function. In our data analysis, we consider the following three popular kernels:

- Linear: $h(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i' \mathbf{x}_j$
- d^{th} Degree Polynomial: $h(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i' \mathbf{x}_j)^d$
- Radial: $h(\mathbf{x}_i, \mathbf{x}_j) = \exp\{-u \|\mathbf{x}_i - \mathbf{x}_j\|^2\}$, where u is a fixed constant.

Now, (25) can be written as

$$\delta_{SVM}(\boldsymbol{\omega}) \equiv \delta_{SVM}(\mathbf{x}, \boldsymbol{\zeta}, \beta_0, \lambda) = \sum_{i=1}^n \zeta_i y_i^* h(\mathbf{x}, \mathbf{x}_i) + \beta_0, \quad (28)$$

And $\hat{\delta}_{SVM}(\mathbf{x}^0, \boldsymbol{\zeta}, \beta_0, \lambda) = \sum_{i=1}^n \hat{\zeta}_i y_i^* h(\mathbf{x}^0, \mathbf{x}_i) + \hat{\beta}_0$.

When implementing this classification method, we use the R package `e1071` (Dimitriadou et al., 2010), to compute $\hat{\zeta}_i$ and $\hat{\beta}_0$ via a quadratic optimization function for a fixed value of λ .

A.4 k-Nearest Neighbors

A.4.1 k-Nearest Neighbors in Covariate Space

The k-nearest neighbors (kNN) classification method makes no assumptions about an underlying model. Using this method, for a point $\{Y^0, \mathbf{x}^0\}$, the closest k points $\{\mathbf{x}_{(r)}, r = 1, \dots, k\}$ to \mathbf{x}^0 are identified, and Y^0 is assigned to the most popular class among the k neighbors, where ties are broken at random. “Distance” here is measured in covariate space, not geographic space, and could be defined using any valid distance metric. In our implementation of the method, we use Euclidean distance so that

$$d_{(r)} = \|\mathbf{x}_{(r)} - \mathbf{x}^0\|.$$

Here, $d_{(i)}$ represents the ordered distances where the minimum is $d_{(1)}$ and the maximum is $d_{(n)}$, and $\mathbf{x}_{(r)}$ are the \mathbf{x}_i corresponding to $d_{(r)}$. Using this measure of distance requires standardization of the variables so that no variable is given more weight than another.

For the binary case, a decision function can then be defined as

$$\delta_{kNN-C}(\omega) \equiv \delta_{kNN-C}(\mathbf{x}^0, \mathbf{x}, \mathbf{Y}) = 2 \sum_{r=1}^k y_{(r)} / k,$$

where $y_{(r)}$ is the y_i associated with $d_{(r)}$. When classifying Y^0 , if $\delta_{kNN-C}(\omega) = 1$, $y_{pred}^0 = 1$ with probability .5 and $y_{pred}^0 = 0$ with probability .5.

A.4.2 k-Nearest Neighbors in Geographic Space

Instead of using covariates to determine proximity as in Section A.4.1, we can also use geographic space. Using this approach, for prediction at location \mathbf{s}^0 , the closest k observed points are identified $\{\mathbf{s}_{(r)}, r = 1, \dots, k\}$, and Y^0 is assigned to the most popular class among its k geographic neighbors. Just as in k-Nearest Neighbors for covariate space, we can use any distance metric, but we again use Euclidean distance so that

$$d_{(r)}^* = \|\mathbf{s}_{(r)} - \mathbf{s}^0\|.$$

The $d_{(i)}^*$ represent ordered distances due to geographical space, and $\mathbf{s}_{(r)}$ are the \mathbf{s}_i corresponding to $d_{(r)}^*$. The decision function for kNN based on geographic space can then be defined as

$$\delta_{kNN-G}(\omega) \equiv \delta_{kNN-G}(\mathbf{s}^0, \mathbf{s}, \mathbf{Y}) = 2 \sum_{r=1}^k y_{(r)}^* / k,$$

where $y_{(r)}^*$ are the y_i associated with $d_{(r)}^*$. When classifying Y^0 , if $\delta_{kNN-G}(\omega) = 1$, we assign $y_{pred}^0 = 1$ with probability .5 and $y_{pred}^0 = 0$ with probability .5.

A.5 Spatial Extensions of Discriminant Analysis

A.5.1 Switzer

Switzer (1980) extend LDA by augmenting the covariates of the focal location with an average of neighboring covariate values. Specifically, let $\mathbf{x}^* = (\mathbf{x}^{0'}, \mathbf{x}^{c'})'$, where

$$\mathbf{x}^c = \frac{1}{n^*} \sum_{j=1}^{n^*} \mathbf{x}_j,$$

where \mathbf{x}_j for $j = 1, \dots, n^*$ are the covariate values of the n^* neighbors of the focal location. In our empirical analysis, we use second-order neighborhood structure, where locations sharing an edge or corner with the focal location are considered neighbors. Then, the decision function is the same as the decision function for LDA (Equation 24), where we replace \mathbf{x}^0 with \mathbf{x}^* .

A.5.2 Mardia

Mardia (1984) uses the same idea as Switzer (1980), but weights the augmented covariates according to a spatial correlation matrix. In this case, for the focal location, we choose a “window” of neighbors which defines n^* neighbors around the focal location. Often, a 3×3 window is used (this corresponds to a second-order neighborhood structure), and we used this in our analysis. We denote the $(n^* + 1) \times \ell$ matrix of covariates belonging to the focal location and its n^* neighbors by

$$\mathbf{X}^* = \begin{bmatrix} \mathbf{x}^{0'} \\ \mathbf{x}'_1 \\ \vdots \\ \mathbf{x}'_{n^*} \end{bmatrix}.$$

The decision function can then be written as

$$\delta_{Mardia}(\omega) \equiv \delta_{Mardia}(\mathbf{X}^*, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_0, \boldsymbol{\Lambda}_1, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) = \frac{S_1}{S_0}$$

where

$$S_j = \log \pi_j - \frac{n^* + 1}{2} \log |\boldsymbol{\Lambda}_j| + \ell(n^* + 1) \log \psi_j - \frac{1}{2\psi_j^2} (G_j - \psi_j^2 \boldsymbol{\mu}_j)' \boldsymbol{\Lambda}_j^{-1} (G_j - \psi_j^2 \boldsymbol{\mu}_j),$$

$$G_j = \mathbf{1}' K^0(\boldsymbol{\theta}_j) \mathbf{X}^*,$$

$$\psi_j^2 = \mathbf{1}' K^0(\boldsymbol{\theta}_j) \mathbf{1},$$

and $\boldsymbol{\mu}_j$ is the $\ell \times 1$ class-specific mean, $\boldsymbol{\Lambda}_j$ is the $\ell \times \ell$ class-specific covariance matrix among the covariates, $K^0(\boldsymbol{\theta}_j)$ is the $(n^* + 1) \times (n^* + 1)$ spatial correlation matrix of the focal location and its n^* neighbors parameterized by the class-specific spatial covariance parameters, $\boldsymbol{\theta}_j$.

For a fixed spatial correlation matrix, Mardia (1984) provides a method to estimate the other parameters. For the training data in class j , let

$$\mathbf{X}_{(j)} = \begin{bmatrix} \mathbf{x}'_1 \\ \vdots \\ \mathbf{x}'_{n_j} \end{bmatrix}$$

be an $n_j \times \ell$ matrix of covariates $\{\mathbf{x}_1, \dots, \mathbf{x}_{n_j}\} = \{\mathbf{x}_i : y_i = j\}$. Then,

$$\begin{aligned}\hat{\boldsymbol{\mu}}_j &= \frac{\mathbf{1}' K(\boldsymbol{\theta}_j)^{-1} \mathbf{X}_{(j)}}{\mathbf{1}' K(\boldsymbol{\theta}_j)^{-1} \mathbf{1}} \\ \hat{\boldsymbol{\Lambda}}_j &= \frac{1}{n_j} (\mathbf{X}_{(j)} - (\mathbf{1} \otimes \hat{\boldsymbol{\mu}}'_j))' K(\boldsymbol{\theta}_j)^{-1} (\mathbf{X}_{(j)} - (\mathbf{1} \otimes \hat{\boldsymbol{\mu}}'_j)),\end{aligned}$$

where $K(\boldsymbol{\theta}_j)$ is the fixed $n_j \times n_j$ spatial correlation matrix of class j . To estimate the spatial correlation, they recommend using covariograms of the covariates in the training data for each class. Finally, as with LDA, we can also assume the same covariance and spatial correlation structure across the classes.

A.5.3 Spatial LDA

Šaltytė Benth and Dučinskas (2005) define a spatio-temporal discriminant analysis approach. Because the data we model is not temporal, we provide a space-only version of their model here.

$$\delta_{SLDA}(\boldsymbol{\omega}) \equiv \delta_{SLDA}(\mathbf{X}, \mathbf{u}^0, \mathbf{B}_1, \mathbf{B}_0, \boldsymbol{\Lambda}) = \left(\mathbf{x}^0 - \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0}{2} \right)' \boldsymbol{\Lambda}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + \log \left(\frac{\pi_1}{\pi_0} \right)$$

where \mathbf{x}^0 are the inputs at the location of interest, $\boldsymbol{\Lambda}$ is the covariance among the covariates, $\boldsymbol{\mu}_j$ is the class-specific mean modeled with a $q \times 1$ vector of regressors, \mathbf{u}^0 , such that

$$\boldsymbol{\mu}_j = \mathbf{B}'_j \mathbf{u}^0,$$

and \mathbf{B}_j is a $q \times \ell$ class-specific matrix of coefficients. Often, \mathbf{u}^0 is just the location coordinates.

To estimate these parameters, let

$$\mathbf{U}_j = \begin{bmatrix} \mathbf{u}'_1 \\ \vdots \\ \mathbf{u}'_{n_j} \end{bmatrix}.$$

Then, for a fixed spatial correlation matrix, $K(\boldsymbol{\theta}_j)$, the maximum likelihood estimators are

$$\begin{aligned}\hat{\mathbf{B}}_j &= (\mathbf{U}'_j K(\boldsymbol{\theta}_j)^{-1} \mathbf{U}_j)^{-1} \mathbf{U}'_j K(\boldsymbol{\theta}_j)^{-1} \mathbf{X}_{(j)}, \\ \hat{\boldsymbol{\Sigma}} &= \frac{1}{n_1 + n_0} \sum_{j=0}^1 (\mathbf{X}_{(j)} - \mathbf{U}_j \hat{\mathbf{B}}_j)' K(\boldsymbol{\theta}_j)^{-1} (\mathbf{X}_{(j)} - \mathbf{U}_j \hat{\mathbf{B}}_j),\end{aligned}$$

where $\mathbf{X}_{(j)}$ is as defined in Section A.5.2. Again we use the variogram to estimate $\boldsymbol{\theta}_j$.

A.6 Press

The final spatial method from Press (1996) uses the class to determine neighborhoods for evaluating the probabilities of the focal point belonging to each class. These probabilities are found by marginalizing over the parameters in the likelihood distribution. Their algorithm is as follows:

1. Using the training data, draw samples from the posterior distributions of the parameters for each class. In our analysis, for class j , we used a multivariate normal distribution allowing for spatial dependence among the covariates. Specifically, we assumed

$$\mathbf{X}_j \sim \mathcal{N}(\mathbf{1} \otimes \boldsymbol{\mu}_j, K(\boldsymbol{\theta}_j) \otimes \boldsymbol{\Lambda}_j),$$

where \mathbf{X}_j is the $\ell n_j \times 1$ vector of covariates in class j , $\boldsymbol{\mu}_j$ is the $\ell \times 1$ class-specific mean, $K(\boldsymbol{\theta}_j) \equiv K(\theta_j)$ is a spatial correlation matrix, and $\boldsymbol{\Lambda}_j$ is the $\ell \times \ell$ class-specific covariance among the covariates. We used the following prior distributions:

$$\begin{aligned}\boldsymbol{\mu}_j &\sim \mathcal{N}(\mathbf{0}, 10000 \mathbf{I}) \\ \theta_j &\sim \text{U}(0, 20) \\ \boldsymbol{\Lambda}_j &\sim \text{InvWishart}(5, \mathbf{I}).\end{aligned}$$

2. Pre-classify each location using a “zero-neighbor Bayesian” classification. Using the samples from the posterior distributions, we computed

$$\begin{aligned}p_j^{0[t]} &= f(\mathbf{x}^0 | \boldsymbol{\mu}_j^{[t]}, \boldsymbol{\Lambda}_j^{[t]}) \\ Y^{0[t]} &= \begin{cases} 1, & \text{if } p_1^{0[t]} > p_0^{0[t]} \\ 0, & \text{otherwise} \end{cases}\end{aligned}$$

where $f(\cdot | \boldsymbol{\mu}_j^{[t]}, \boldsymbol{\Lambda}_j^{[t]})$ is the normal distribution with the t th draw of $\boldsymbol{\mu}_j$ as the mean, and the t th draw of $\boldsymbol{\Lambda}_j$ as the covariance. Y^0 is then pre-classified to the most frequent class among the $Y^{0[t]}$. Denote the set of all pre-classified observations by $\mathbf{y}^{pc} = \{y_1^{pc}, \dots, y_n^{pc}\}$.

3. For each location, identify a homogeneous neighborhood using a degree of homogeneity measure to compare possible neighborhoods. In our analysis, we compared four possible directional neighborhoods – north, south, east, and west – where the neighborhood made up the 8 locations in the named direction along with the focal location. The neighborhood for class j is the neighborhood with the most locations pre-classified to class j . If there were multiple neighborhoods with the same number of locations in class j , we randomly selected one. Call the selected neighborhood for class j of the focal location N_j^0 .
4. For each location, compute

$$\Delta_j^0 \propto \pi_j P(\mathbf{X}_j^0 | y^{0pc} = j, y_1^{pc} = j, \dots, y_{n_j^*}^{pc} = j),$$

where y^{0pc} is the pre-classified value of the focal location, $y_1^{pc}, \dots, y_{n_j^*}^{pc}$ correspond to the n_j^* neighbors pre-classified to j in N_j^0 and

$$\mathbf{X}_j^0 = \begin{bmatrix} \mathbf{x}^0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{n_j^*} \end{bmatrix}.$$

In our analysis, we computed

$$\Delta_j^{0[t]} = \pi_j f(\mathbf{X}_j^0 | \boldsymbol{\mu}_j^{[t]}, \theta_j^{[t]}, \boldsymbol{\Lambda}_j^{[t]})$$

$$\hat{Y}^{0[t]} = \begin{cases} 1, & \text{if } \Delta_{i1}^{[t]} > \Delta_{i0}^{[t]} \\ 0, & \text{otherwise} \end{cases}$$

where $f(\cdot | \boldsymbol{\mu}_j^{[t]}, \theta_j^{[t]}, \boldsymbol{\Lambda}_j^{[t]}) = \mathcal{N}(\mathbf{1} \otimes \boldsymbol{\mu}_j^{[t]}, K(\theta_j^{[t]}) \otimes \boldsymbol{\Lambda}_j^{[t]})$. Finally, y_{pred}^0 is classified to be the most frequent class among the $\hat{Y}^{0[t]}$.

We can write the decision function as

$$\delta_{Press}(\boldsymbol{\omega}) \equiv \delta_{Press}(\mathbf{X}_0^0, \mathbf{X}_1^0, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \theta_0, \theta_1, \boldsymbol{\Lambda}_0, \boldsymbol{\Lambda}_1) = \frac{\Delta_1^0}{\Delta_0^0},$$

where $\Delta_j^0 = \pi_j f(\mathbf{X}_j^0 | \boldsymbol{\mu}_j, \theta_j, \boldsymbol{\Lambda}_j)$. Note that the algorithm described above corresponds to the posterior predictive classifier (Section 2.2.1) based on this decision function because it marginalizes over the posterior distributions of parameters $\boldsymbol{\mu}_j$, θ_j , and $\boldsymbol{\Lambda}_j$.

For more details and options for other approaches to pre-classification and selecting a directional neighborhood, we refer the reader to Press (1996).

B Model-fitting Algorithms for the SGLM/SGLMM

B.1 Data Augmentation Algorithms

In Section 2.1, ϵ_i in equation (6) has a fixed variance of 1. This is for identifiability purposes and also creates the probit link. However, for model-fitting purposes, instead of fixing the variance parameter to 1, we can use and marginalize over a variance parameter (for examples in the Bayesian probit model, see Imai and van Dyk, 2005; Berrett and Calder, 2012). Berrett and Calder (2012) describe a model-fitting algorithm for the probit SGLM. Here we show how this algorithm can be extended to create a data augmentation model-fitting algorithm for the SGLMM. We briefly review the model notation and model-fitting algorithm here and then provide an adapted algorithm for the probit SGLMM. In both algorithms, we make use of a non-identifiable variance parameter of the latent variable to facilitate mixing of the MCMC. Within this appendix, we use the $\tilde{\cdot}$ notation to represent unidentifiable parameters. For both algorithms, we used what Berrett and Calder (2012) call the Non-collapsed Marginal-Scheme 1 Algorithm.

Equation (5) in Section 2.1 describes the spatial covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ to be a scalar, θ_1 , times a spatial dependence matrix $K(\boldsymbol{\theta}_2)$. For the SGLM, θ_1 is the non-identifiable parameter which Berrett and Calder (2012) make use of as a working parameter. For consistency of the $\tilde{\cdot}$ notation, let $\tilde{K}(\boldsymbol{\theta}) = \sigma^2 K(\boldsymbol{\theta})$, where $\tilde{K}(\boldsymbol{\theta}) = \boldsymbol{\Sigma}(\boldsymbol{\theta})$ is the spatial covariance matrix, $\sigma^2 = \theta_1$ is a non-identifiable scalar used as the working parameter of the algorithm, and $K(\boldsymbol{\theta})$ is the identifiable spatial dependence matrix parameterized by a single parameter, $\boldsymbol{\theta}$.

The data augmentation model-fitting algorithm for the SGLM is then

Step 1: Sample $\sigma_{temp}^2 \sim \pi(\sigma^2)$
 Sample $\tilde{\mathbf{Z}}|\mathbf{Y}, \boldsymbol{\beta}, \theta, \sigma_{temp}^2$
 Set $\mathbf{Z} = \tilde{\mathbf{Z}}/\sigma_{temp}$
Step 2: Sample $(\sigma^2, \boldsymbol{\beta})|\tilde{\mathbf{Z}}, \mathbf{Y}, \theta$
 Set $\boldsymbol{\beta} = \tilde{\boldsymbol{\beta}}/\sigma$
Step 3: Sample $\theta|\tilde{\mathbf{Z}}, \mathbf{Y}, \tilde{\boldsymbol{\beta}}, \sigma^2$.

For more details, see Berrett and Calder (2012).

For the SGLMM, we can build on this model-fitting algorithm by adding an additional step to sample the additional parameter. Recalling that $\epsilon_i \sim \mathcal{N}(0, 1)$, we can write

$$(\boldsymbol{\nu} + \tilde{\boldsymbol{\epsilon}}) \sim \mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{\Sigma}}^*(\boldsymbol{\theta})),$$

where

$$\tilde{\boldsymbol{\Sigma}}^*(\boldsymbol{\theta}) = \mathbf{I} + \boldsymbol{\Sigma}(\boldsymbol{\theta}) = \mathbf{I} + \sigma^2 K(\boldsymbol{\theta}),$$

where $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is defined as in the SGLM algorithm above. Letting $\gamma^2 = 1 + \sigma^2$ and $\kappa = \sigma^2/\gamma^2$, we obtain

$$\tilde{\boldsymbol{\Sigma}}^*(\boldsymbol{\theta}) = \gamma^2 ((1 - \kappa)\mathbf{I} + \kappa K(\boldsymbol{\theta}_2)).$$

This is simply a scalar times a spatial dependence matrix, just as in the model-fitting algorithm for the SGLM. Notice that κ is bounded by 0 and 1 and that if $\kappa = 1$, $\tilde{\boldsymbol{\Sigma}}^*(\boldsymbol{\theta}) = \sigma^2 K(\boldsymbol{\theta})$, or the spatial covariance matrix of the SGLM. Therefore, to adapt the SGLM data augmentation algorithm to the SGLMM, we let γ^2 be the non-identifiable working parameter and add a step to sample κ .

Here we provide the model-fitting algorithm and the full conditional distributions for the SGLMM. We use priors $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_{\boldsymbol{\beta}})$, $\gamma^2 \sim b_{\gamma}(\chi_{a_{\gamma}}^2)^{-1}$, $\theta \sim \pi(\theta)$, and $\kappa \sim \text{U}(0, 1)$. We use superscript $[t]$ to denote the value of a parameter at the t th iteration of the algorithm. Let $\tilde{\boldsymbol{\Sigma}}^*(\boldsymbol{\theta}) = \gamma^2 \boldsymbol{\Sigma}^*(\boldsymbol{\theta})$, where $\boldsymbol{\Sigma}^*(\boldsymbol{\theta}) = (1 - \kappa)\mathbf{I} + \kappa K(\boldsymbol{\theta})$.

Step 1: Sample $\mathbf{Z}^{[t]}$ from $\mathbf{Z}|\mathbf{Y}, \boldsymbol{\beta}^{[t-1]}, \theta^{[t-1]}, \kappa^{[t-1]}$:

Draw $\gamma_{temp}^2 \sim \pi(\gamma^2)$.

For $i = 1, \dots, n$, define $\mathbf{Z}_{-i}^{[t-1]} = (Z_1^{[t-1]}, \dots, Z_{i-1}^{[t-1]}, Z_{i+1}^{[t-1]}, \dots, Z_n^{[t-1]})'$ and sample \tilde{Z}_i from

$$\tilde{Z}_i|\mathbf{Y}, \mathbf{Z}_{-i}^{[t-1]}, \boldsymbol{\beta}^{[t-1]}, \theta^{[t-1]}, \kappa^{[t-1]}, \gamma_{temp}^2 \sim \begin{cases} TN(\mu_{\tilde{Z}_i}, \tau_{\tilde{Z}_i}^2, 0, \infty), & \text{if } Y_i = 1 \\ TN(\mu_{\tilde{Z}_i}, \tau_{\tilde{Z}_i}^2, -\infty, 0), & \text{if } Y_i = 0 \end{cases},$$

where $TN(\mu_{\tilde{Z}_i}, \tau_{\tilde{Z}_i}^2, \ell, u)$ is a truncated normal distribution with lower and upper bounds ℓ and u , respectively, and mean and variance

$$\begin{aligned} \mu_{\tilde{Z}_i} &= \gamma_{temp} \mathbf{x}_i' \boldsymbol{\beta}^{[t-1]} + [\boldsymbol{\Sigma}^*(\boldsymbol{\theta}^{[t]})]_{i,-i} \left([\boldsymbol{\Sigma}^*(\boldsymbol{\theta}^{[t]})]_{-i,-i} \right)^{-1} \gamma_{temp} \left(\mathbf{Z}_{-i}^{[t-1]} - \mathbf{X}_{-i} \boldsymbol{\beta}^{[t-1]} \right) \\ \tau_{\tilde{Z}_i}^2 &= \gamma_{temp}^2 \left([\boldsymbol{\Sigma}^*(\boldsymbol{\theta}^{[t]})]_{i,i} - [\boldsymbol{\Sigma}^*(\boldsymbol{\theta}^{[t]})]_{i,-i} \left([\boldsymbol{\Sigma}^*(\boldsymbol{\theta}^{[t]})]_{-i,-i} \right)^{-1} [\boldsymbol{\Sigma}^*(\boldsymbol{\theta}^{[t]})]_{-i,i} \right). \end{aligned}$$

Set $Z_i^{[t]} = \tilde{Z}_i^{[t]}/\gamma_{temp}$.

For an unobserved location, sample

$$\tilde{Z}^0 | \mathbf{x}^0, \mathbf{Y}, \mathbf{Z}^{[t]}, \boldsymbol{\beta}^{[t-1]}, \theta^{[t-1]}, \kappa^{[t-1]}, \gamma_{temp}^2 \sim \mathcal{N}(\gamma_{temp} \mu_{Z^0}, \gamma_{temp}^2 \sigma_{Z^0}^2),$$

where μ_{Z^0} and $\sigma_{Z^0}^2$ are defined in equations (13) and (14), and set $Z^{0[t]} = \tilde{Z}^0 / \gamma_{temp}$.

Step 2: Sample $(\gamma^2)^{[t]}, \boldsymbol{\beta}^{[t]}$ from $\gamma^2, \boldsymbol{\beta} | \mathbf{Y}, \tilde{\mathbf{Z}}^{[t]}, \theta^{[t-1]}, \kappa^{[t-1]}$:
Sample

$$(\gamma^2)^{[t]} \sim \left((\tilde{\mathbf{Z}} - \mathbf{X}\hat{\boldsymbol{\beta}})' \left(\boldsymbol{\Sigma}^*(\theta^{[t-1]}) \right)^{-1} (\tilde{\mathbf{Z}} - \mathbf{X}\hat{\boldsymbol{\beta}}) + b_\gamma^2 + \hat{\boldsymbol{\beta}}' \mathbf{V}_\beta^{-1} \hat{\boldsymbol{\beta}} \right) (\chi_{n+a_\gamma}^2)^{-1},$$

where $\hat{\boldsymbol{\beta}} = \left(\mathbf{X}' \left(\boldsymbol{\Sigma}^*(\theta^{[t-1]}) \right)^{-1} \mathbf{X} + \mathbf{V}_\beta^{-1} \right)^{-1} \mathbf{X}' \left(\boldsymbol{\Sigma}^*(\theta^{[t-1]}) \right)^{-1} \tilde{\mathbf{Z}}^{[t]}$. Sample

$$\tilde{\boldsymbol{\beta}} \sim \mathcal{N} \left(\hat{\boldsymbol{\beta}}, (\gamma^2)^{[t]} \left(\mathbf{X} \left(\boldsymbol{\Sigma}^*(\theta^{[t-1]}) \right)^{-1} \mathbf{X} + \mathbf{V}_\beta^{-1} \right)^{-1} \right).$$

Set $\boldsymbol{\beta}^{[t]} = \tilde{\boldsymbol{\beta}} / \gamma^{[t]}$.

Step 3: Sample $\theta^{[t]}$ from $\theta | \mathbf{Y}, \tilde{\mathbf{Z}}^{[t]}, \tilde{\boldsymbol{\beta}}^{[t]}, (\gamma^2)^{[t]}, \kappa^{[t-1]}$ via a random walk Metropolis step:
Sample a proposal value θ_{prop} from a proposal distribution. We used a normal distribution, $\mathcal{N}(\theta | \theta^{[t-1]}, \tau_\theta^2)$, where τ_θ^2 is the fixed variance of the proposal distribution. Define

$$\theta^{[t]} = \begin{cases} \theta_{prop} & \text{with probability } c(\theta^{[t-1]}, \theta_{prop}) \\ \theta^{[t-1]} & \text{with probability } 1 - c(\theta^{[t-1]}, \theta_{prop}) \end{cases}$$

where

$$c(\theta^{[t-1]}, \theta_{prop}) = \min \left\{ \frac{\pi(\theta_{prop} | \mathbf{Y}, \tilde{\mathbf{Z}}, \tilde{\boldsymbol{\beta}}, (\gamma^2)^{[t]}, \kappa^{[t-1]})}{\pi(\theta^{[t-1]} | \mathbf{Y}, \tilde{\mathbf{Z}}, \tilde{\boldsymbol{\beta}}, (\gamma^2)^{[t]}, \kappa^{[t-1]})}, 1 \right\}.$$

The posterior distribution of θ in the acceptance probability is

$$\pi(\theta_{prop} | \mathbf{Y}, \tilde{\mathbf{Z}}, \tilde{\boldsymbol{\beta}}, (\gamma^2)^{[t]}, \kappa^{[t-1]}) \propto \phi(\tilde{\mathbf{Z}}; \mathbf{X}\tilde{\boldsymbol{\beta}}, \gamma^2 \boldsymbol{\Sigma}^*(\theta)) \pi(\theta),$$

where $\phi(\cdot)$ is the multivariate normal density function.

Step 4: Sample $\kappa^{[t]}$ from $\kappa | \mathbf{Y}, \tilde{\mathbf{Z}}^{[t]}, \tilde{\boldsymbol{\beta}}^{[t]}, (\gamma^2)^{[t]}, \theta^{[t]}$ via a random walk Metropolis step:
Sample a proposal value κ_{prop} from a proposal distribution. We used a normal distribution, $\mathcal{N}(\kappa | \kappa^{[t-1]}, \tau_\kappa^2)$, where τ_κ^2 is the fixed variance of the proposal distribution. Define

$$\kappa^{[t]} = \begin{cases} \kappa_{prop} & \text{with probability } c(\kappa^{[t-1]}, \kappa_{prop}) \\ \kappa^{[t-1]} & \text{with probability } 1 - c(\kappa^{[t-1]}, \kappa_{prop}) \end{cases}$$

where

$$c(\kappa^{[t-1]}, \kappa_{prop}) = \min \left\{ \frac{\pi(\kappa_{prop} | \mathbf{Y}, \tilde{\mathbf{Z}}, \tilde{\boldsymbol{\beta}}, (\gamma^2)^{[t]}, \theta^{[t]})}{\pi(\kappa^{[t-1]} | \mathbf{Y}, \tilde{\mathbf{Z}}, \tilde{\boldsymbol{\beta}}, (\gamma^2)^{[t]}, \theta^{[t]})}, 1 \right\}.$$

The posterior distribution of κ in the acceptance probability is

$$\pi(\kappa_{prop} | \mathbf{Y}, \tilde{\mathbf{Z}}, \tilde{\boldsymbol{\beta}}, (\gamma^2)^{[t]}, \theta^{[t]}) \propto \phi(\tilde{\mathbf{Z}}; \mathbf{X}\tilde{\boldsymbol{\beta}}, \gamma^2 \boldsymbol{\Sigma}^*(\theta)) \pi(\kappa),$$

where $\phi(\cdot)$ is the multivariate normal density function.

B.2 Low-rank Probit SGLMM

In this section, we provide an overview and a data augmentation algorithm for fitting the Hughes and Haran (2013) probit version of their low-rank SGLMM.

Originally proposed by Reich et al. (2006), Hughes and Haran decompose the ν_i 's in equation (3) into two pieces: one collinear with \mathbf{X} (this component is discarded) and one orthogonal to \mathbf{X} . Hughes and Haran use the Moran operator to do this. Letting \mathbf{A} be the binary neighborhood matrix, where the ij th element is 1 if locations i and j are neighbors, and 0 otherwise, the Moran operator is defined to be

$$\mathbf{M} = \left(\mathbf{I} - \mathbf{X} (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}' \right) \mathbf{A} \left(\mathbf{I} - \mathbf{X} (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}' \right).$$

Let Ψ be the $n \times r$ eigenvectors of \mathbf{M} corresponding to the r largest eigenvalues. For gridded data, Hughes and Haran propose letting r be equal to 10 percent of the eigenvectors. We can then adjust equation (6) to reflect this low rank version of the SGLMM:

$$Z_i = \mathbf{x}_i' \beta_1 + \Psi_i \beta_2 + \epsilon_i,$$

where β_1 is the $\ell \times 1$ vector of coefficients corresponding to the ℓ covariates and β_2 is the $r \times 1$ vector of coefficients corresponding to Ψ_i , the i th row of the the selected r Moran operator eigenvectors.

In this approach, the spatial dependence is captured as additional covariates and coefficients. Letting

$$\mathbf{X}_H = [\mathbf{X} \quad \Psi]$$

and

$$\beta_H = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix},$$

we can write the spatial latent variable Bayesian probit model in the form of a non-spatial latent variable Bayesian probit model:

$$Z_i = \mathbf{x}_{Hi}' \beta_H + \epsilon_i,$$

where \mathbf{x}_{Hi} is the $(\ell + r) \times 1$ vector corresponding to the i th row of \mathbf{X}_H and $\epsilon_i \sim \mathcal{N}(0, \gamma^2)$. Setting prior distributions to be

$$\begin{aligned} \beta_H &\sim \mathcal{N}(\mathbf{0}, \mathbf{V}_H), \\ \gamma^2 &\sim b_\gamma(\chi_{a_\gamma}^2)^{-1}, \end{aligned}$$

we use the following data augmentation model-fitting algorithm:

Step 1: Sample $\mathbf{Z}^{[t]}$ from $\mathbf{Z}|\mathbf{Y}, \beta_H^{[t-1]}$:

Draw $\gamma_{temp}^2 \sim \pi(\gamma^2)$.

For $i = 1, \dots, n$, sample \tilde{Z}_i from

$$\tilde{Z}_i | \mathbf{Y}, \beta_H^{[t-1]} \sim \begin{cases} TN(\gamma_{temp} \mathbf{x}_{Hi}' \beta_H, \gamma_{temp}^2, 0, \infty), & \text{if } Y_i = 1 \\ TN(\gamma_{temp} \mathbf{x}_{Hi}' \beta_H, \gamma_{temp}^2, -\infty, 0), & \text{if } Y_i = 0 \end{cases}.$$

Set $Z_i^{[t]} = \tilde{Z}_i / \gamma_{temp}$.

For an unobserved location, sample \tilde{Z}^0 from $\tilde{Z}^0 | \beta_H^{[t-1]} \sim \mathcal{N}(\gamma_{temp} \mathbf{x}_H^{0'} \beta_H, \gamma_{temp}^2)$ and set $Z^{0[t]} = \tilde{Z}^0 / \gamma_{temp}$.

Step 2: Sample $(\gamma^2)^{[t]}, \beta_H^{[t]}$ from $\gamma^2, \beta_H | \tilde{\mathbf{Z}}, \mathbf{Y}$:

Sample

$$(\gamma^2)^{[t]} \sim \left((\tilde{\mathbf{Z}} - \mathbf{X}_H \hat{\beta}_H)' (\tilde{\mathbf{Z}} - \mathbf{X}_H \hat{\beta}_H) + b_\gamma^2 + \hat{\beta}_H' \mathbf{V}_H^{-1} \hat{\beta}_H \right) (\chi_{n+a_\gamma}^2)^{-1},$$

where $\hat{\beta}_H = (\mathbf{X}_H' \mathbf{X}_H + \mathbf{V}_H^{-1})^{-1} \mathbf{X}_H' \tilde{\mathbf{Z}}^{[t]}$.

Sample

$$\tilde{\beta}_H \sim \mathcal{N} \left(\hat{\beta}_H, (\gamma^2)^{[t]} (\mathbf{X}_H \mathbf{X}_H + \mathbf{V}_H^{-1})^{-1} \right).$$

Set $\beta_H^{[t]} = \tilde{\beta}_H / \gamma^{[t]}$.

C Simulation Study Classification Error Rates

Error rates for the simulation study in Section 3.2 are provided in the tables below.

One-at-a-Time Training							
κ		0.25		0.5		1	
Model Fit		SGLM	SGLMM	SGLM	SGLMM	SGLM	SGLMM
Linear Component Intercept	Data Set						
	1	0.3859	0.3859	0.4048	0.4118	0.2177	0.2381
	2	0.3232	0.3502	0.3795	0.3861	0.2287	0.2253
	3	0.3432	0.3432	0.2466	0.274	0.2905	0.3041
Simple-1	1	0.3345	0.3412	0.2774	0.2774	0.1347	0.1481
	2	0.3255	0.3255	0.3028	0.3134	0.1259	0.1399
	3	0.354	0.354	0.3428	0.3498	0.1443	0.1512
Simple-2	1	0.3559	0.3593	0.2891	0.2993	0.1103	0.1345
	2	0.323	0.3127	0.2432	0.25	0.1424	0.1458
	3	0.3017	0.3051	0.3389	0.3423	0.1577	0.1846
Multiple	1	0.3172	0.331	0.2449	0.2585	0.1513	0.1612
	2	0.2902	0.3217	0.2517	0.2586	0.1533	0.1672
	3	0.2437	0.2616	0.2457	0.2803	0.1325	0.1358
Confounded	1	0.2399	0.2601	0.2068	0.2271	0.0993	0.0925
	2	0.2295	0.2226	0.2990	0.3156	0.0976	0.1077
	3	0.2748	0.2682	0.2517	0.2653	0.1559	0.1661

Joint Training							
κ Model Fit		0.25		0.5		1	
		SGLM	SGLMM	SGLM	SGLMM	SGLM	SGLMM
Linear Component Intercept	Data Set						
	1	0.3859	0.3859	0.4048	0.4048	0.2789	0.2789
	2	0.4411	0.4411	0.3861	0.3861	0.5119	0.5154
	3	0.3432	0.3432	0.2774	0.2774	0.4595	0.4561
Simple-1	1	0.3446	0.348	0.3082	0.3151	0.2189	0.2256
	2	0.3255	0.3154	0.3204	0.3204	0.2483	0.2483
	3	0.3643	0.3608	0.3463	0.3463	0.2337	0.2337
Simple-2	1	0.3525	0.3559	0.3163	0.3129	0.1862	0.1793
	2	0.3127	0.3127	0.2603	0.2603	0.2407	0.2407
	3	0.3119	0.3017	0.3456	0.3456	0.2114	0.2114
Multiple	1	0.3241	0.3207	0.2619	0.2619	0.1875	0.1809
	2	0.3182	0.3252	0.2586	0.2586	0.2125	0.2125
	3	0.2688	0.276	0.2907	0.2907	0.2053	0.1987
Confounded	1	0.2601	0.2601	0.2712	0.2712	0.1507	0.1507
	2	0.2226	0.2295	0.3223	0.3156	0.1313	0.1313
	3	0.2616	0.2682	0.2721	0.2721	0.1729	0.1661

Test							
κ Model Fit		0.25		0.5		1	
		SGLM	SGLMM	SGLM	SGLMM	SGLM	SGLMM
Linear Component Intercept	Data Set						
	1	0.4314	0.4314	0.4054	0.4324	0.2925	0.3302
	2	0.3981	0.3981	0.3711	0.3711	0.3084	0.2991
	3	0.3196	0.3196	0.2222	0.2407	0.2981	0.2981
Simple-1	1	0.4712	0.4327	0.3704	0.3611	0.1553	0.1553
	2	0.3137	0.3235	0.3707	0.3621	0.1842	0.1842
	3	0.422	0.422	0.265	0.265	0.1101	0.1284
Simple-2	1	0.2952	0.2952	0.3019	0.3208	0.1545	0.1727
	2	0.2936	0.3028	0.25	0.2593	0.1905	0.1905
	3	0.2857	0.2857	0.2745	0.2745	0.1667	0.1667
Multiple	1	0.3273	0.3182	0.2547	0.2642	0.1771	0.1875
	2	0.307	0.3333	0.2727	0.2818	0.1416	0.1327
	3	0.3802	0.3967	0.3604	0.3423	0.1122	0.1122
Confounded	1	0.3077	0.3269	0.3048	0.3143	0.2222	0.2500
	2	0.3056	0.3148	0.1717	0.1919	0.1262	0.1359
	3	0.3367	0.3265	0.2453	0.2453	0.2000	0.1905